School of Science
Department of Mathematics

Ground Elevation Models and Land Cover Classifiers for Decimetre Resolution Urban Monitoring

Kassel Liam Hingee

This thesis is presented for the Degree of
Master of Philosophy (Mathematics)
of
Curtin University

July 2013

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made.
This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Kassel Liam Hingee

"The truth is rarely pure and never simple. Modern life would be very tedious if it were either"

- Algernon, The Importance of Being Ernest by Oscar Wilde

Acknowledgements

There is little doubt that this thesis would not have been possible without the assistance and forethought of many people.

My two supervisors, Lou Caccetta and Peter Caccetta, provided excellent assistance and cunning advice throughout the project. I am especially grateful to Peter for his ground filter algorithm, inflows, and his role in starting and driving the Urban Monitor project.

Many people were involved in acquiring and pre-processing the data used in this project. Of particular note were Xiaoliang Wu and Simon Collings. Wu developed the algorithms and programs that generated the DSMs and orthorectified the spectral data, and Simon calibrated the spectral data to ground reflectance. Without their creativity and attention to quality neither the GEM algorithm nor the land cover classifiers would have been successful.

Drew Devereux, Tony Traylen and Zheng-Shu Zhou all helped with the visual inspection of the entire region (chapter 4). They spent much of their valuable time on the task and coped well with my changes to the procedures.

Drew Devereux was also pivotable to the GEM generation (Drew ran the GEM algorithm on the iVEC supercomputing facility) and developed a rasterising program that was used to correct GEM errors.

Many people have indirectly made my research easier. My officemates, primarily Sam Burnham, created an enjoyable and positive working environment. My family, friends and house mates supported me and understood when I was too busy to interact with them.

I would also like to thank the Commonwealth Scientific and Industrial Research Organisation for their support in funding me and providing me with many resources.

Abstract

This thesis presents methods that enable the generation of quantitative environmental indicators for remotely monitoring urban regions. Its contributions are a new algorithm for the generation of ground elevation models and significant research into the feature extraction method canonical variate analysis with rational polynomials (CVAR).

Many organisations desire to measure, analyse and comprehend urban regions to assist with a multitude of issues, including significant changes such as massive population growth (it is predicted that the next 40 years of global population growth will occur entirely in cities, with the population of rural areas *declining* by 0.3 billion - by the year 2050 two thirds of the worlds population will reside in cities [United Nations. World Urbanization Prospects, The 2011 Revision: Highlights. United Nations, New York, USA, 2012]). Remote sensing is an invaluable tool for these organisations due to the large diversity and density of both natural (e.g. trees, lakes) and built (e.g. houses) objects, and the importance of their spatial relationships.

Often sub-metre resolution data is required for urban remote sensing due to the size and complexity of urban objects. Broad region analyses at this resolution have rarely been performed and many of the methods that work well on small areas, are too slow or inappropriate for application to entire cities.

This project used colour infrared (CIR) aerial photography that has been stereo-matched to create a digital surface model (DSM) and a calibrated orthomosaick at $20cm$ resolution over a $9600km^2$ region surrounding Perth, Australia. Digital aerial photography is an ideal source due to its regular capture by governments, its high resolution and its ability to simultaneously capture elevation information (through stereo-matching).

Algorithms were investigated that addressed the calculation of bare earth

surface elevations and the classification of land covers for the entire region. These algorithms are essential for the generation of environmental indicators and many other tasks such as building change detection, virtual city models and heat island detection.

A hybrid algorithm that combined a morphological filter and a surface fitting based filter was developed to estimate the elevation of the ground from the DSM. The surfaces generated achieved 90% accuracy compared with manually interpreted points and were shown to outperform a commercially available algorithm.

For classification, a generalisation of canonical variate analysis (CVA) that produced terrain-illumination resistant discriminant functions was investigated. This generalisation, called *canonical variate analysis with rational polynomials* (CVAR), generated rational polynomial functions that were capable of discriminating multiple classes over at least $16km^2$ areas using simple thresholds. The investigation also revealed some techniques that would facilitate the application of CVAR discrimination functions to larger regions. An NDVI-based vegetation/non-vegetation classifier was also investigated for city-wide classifications.

The investigations into ground elevation generation and land cover classifiers are presented in chapters 2 and 3 respectively. The techniques used to apply these methods to the full $9600km^2$ region are described in chapter 4.

Contents

Acknowledgements **vii**

Abstract **ix**

1 Introduction **1**

1.1 Urban Monitoring and Remote Sensing 3

1.2 CSIRO's Urban Monitor Project 6

1.3 Contributions 7

1.3.1 Ground Elevation Model Generation 7

1.3.2 Classification 8

2 Creation of Ground Elevation Models **11**

2.1 Algorithm Overview 14

2.2 Filtering the DSM into candidate ground points using a segmentation/morphological based algorithm 16

2.3 Multi-resolution Surface Fitting 19

2.3.1 Terzopoulos' Multigrid Algorithm 20

2.3.2 The Cascadic Algorithm 23

2.3.3 Surface Fitting with Discontinuities 31

2.4 Outlier Detection using Surface Fitting 32

2.4.1 The Parameters Used for Outlier Detection 33

2.5 Results of the Ground Elevation Model Algorithm 37

2.5.1 Time Required 40

2.5.2 Accuracy . . . 41
2.5.3 Comparison to an Inpho match-T GEM . . . 42
2.6 Conclusion and Discussion . . . 45

3 Classification **47**
3.1 Classification by Canonical Variate Analysis with Rational Polynomials . . . 49
3.1.1 CVAR Details . . . 51
3.1.2 Experiments in Multiclass Classifications . . . 53
3.1.3 Comparison to CVA and CVAR with Independent Denominators . . . 71
3.1.4 CVAR Topographic Illumination Normalisation . . . 73
3.1.5 Brute Force Search of Starting Simplex . . . 74
3.1.6 Guidelines to the Application of CVAR . . . 77
3.1.7 Discussion . . . 80
3.2 Vegetation Classification using NDVI . . . 81
3.2.1 Methods . . . 84
3.2.2 Results and Discussion . . . 86
3.3 Conclusion and Future Work . . . 88

4 Application to Full Urban Monitor Region **91**
4.1 Methods . . . 92
4.1.1 Generating the Candidate Ground Points, GEM and Roof Mask . . . 92
4.1.2 Generating the Remaining Maps . . . 93
4.1.3 Distributed Manual Inspection and Correction of Products 94
4.2 Results and Discussion . . . 97
4.2.1 Summary Vegetation Statistics . . . 97
4.2.2 Evaluation of Distributed Inspection Methods . . . 98
4.2.3 GEM . . . 98

4.2.4 Vegetation Mask 99

5 Conclusion and Future Work 103

5.1 Land Cover Classification 104

5.2 Ground Elevation Model 105

A Uniqueness and Existence of the Discrete Surface Fitting Problem 107

B Experiments for the generation of ground elevation models 109

B.1 The Number of Levels and the Behaviour of a Gauss-Seidel Relaxation 109

B.1.1 Number of Levels 110

B.1.2 Effect of a Relaxation 110

B.2 Outlier Detection Parameter Search 113

B.2.1 Darling Ranges 113

B.2.2 Perth Coastal Plain 115

C Detailed description of the GEM generation implementation 121

C.1 Implementation of inflows 121

C.2 Implementation of Terzopoulos 122

C.2.1 cascadicMultiresFit 123

C.2.2 removeRoughControlPointswDiscons 125

D Functions and Programs for Land Cover Classifications 127

D.1 Implementation of CVAR 127

D.1.1 Components for CVAR 127

D.1.2 The CVAR function 130

D.2 Accuracy Analysis of Classifications 131

E Appendix for Application to Full Urban Monitor Region 133

E.1 Detection of Errors 133

E.2 Execution of Programs for Data Creation 134
E.3 Description of programs: what they do, and how to use them . . . 139

Bibliography **147**

Chapter 1

Introduction

Monitoring of our surroundings is constantly occurring across many disciplines and organisations using a multitude of methods from direct measurement (e.g. pH meters, thermometers, GPS), manual observation (e.g. bird counts) and proximal sensing (e.g. ground-based radiometers) to remote sensing (airborne and satellite sensors). Some phenomena can be captured comprehensively on a regular basis using direct or proximal sensing but many other phenomena can only be sampled with these techniques (e.g. many fauna and flora populations).

These ground-based techniques are unable to effectively map objects at regional or larger contexts and are impractical for quick reporting and broad scale monitoring[75]. The strength of remotely sensed aerial and satellite data is its ability to supply spatial and quantitative information over broad areas and unrestricted by artificial administrative boundaries [88]; whilst remote sensing is not capable of recording the same detail as in situ measurements, it is able to comprehensively capture other information, often from a new, complimentary perspective.

Remotely sensed data can provide maps of land cover and land use changes. Land use change is considered the most severe driver of change in biodiversity ([79] in [80]) and probably has the largest global impact on biodiversity of any human induced change [98]. Structural vegetation characteristics derived from this land cover data may also be used in the analysis and modelling of dynamic environmental systems such as energy balance, biogeochemical cycles and hydrological cycles [27].

This research focused on remote sensing using digital aerial photography.

Many uses of this data source still rely on manual interpretation [44] and this is likely to continue due to the complexity of our environment. However the time consuming and subjective nature of manual interpretation is causing a growing desire for greater automation.

The initial goal of automated processing is typically to transform the data to assist with manual interpretation or to generate land cover and land cover change classifications. Simple environmental indicators such as the total area of wetlands or forests can easily be generated from these classifications and their changes tracked over multiple dates.

Methods for generating land cover classifications vary wildly. Lu and Weng [64] nicely categorised most of the algorithms with differences including the need for training data, the use of contextual data (spatially neighbouring pixel values) and whether assumptions on the distribution of the data were made. The formats of the outputs also vary with some classifiers assigning a single class or multiple classes (fuzzy or sub-pixel classifiers) to each pixel, or grouping pixels into regions and then classifying (known as object-oriented classifiers)[64]. Many algorithms also use multiple sources of remote sensing data and exploit pre-existing data in geographic information systems (GIS).

Many methods for change detection have been proposed in both remote sensing and the wider image analysis community (see surveys [31, 63, 74, 82]). Image differencing, principle component analysis and post-classification comparisons were the most common techniques by 2004, although the visual interpretation of multidate composite images remained a very common and relatively powerful method [63]. Methods that exploit many dates of data (e.g. temporal modelling/trajectories) are especially powerful with the ability to considerably improve the accuracy of classifications (e.g. [53, 40]) and detect subtle changes [31], but are rarer due to the difficulty in obtaining long time series at appropriate resolutions [31].

The more sophisticated applications such as salinity predictions [14], the impact of bores on tree death [24], bird travel path predictions [70], and model calibration and validation [99] build upon these single-date and multi-date classifiers with the help of statistics, in situ knowledge, modelling and GIS data.

Vegetation indices (quantitative measurements indicating the vigour of vegetation [7]) are one of the simplest tools available, enabling the evaluation of veg-

etative cover density [7] and aiding both manual interpretation and land cover classification.

The sensor types widely used are lidar, radar, multispectral cameras and hyperspectral cameras. The ground sample distances (GSD, the distance on the ground associated with the width of a single pixel) range from centimetres (airborne sensors), tens of centimetres (airborne sensors and new satellites such as Quickbird [101]) to many metres and larger.

Active systems such as lidar and radar emit electromagnetic radiation and detect reflections. Both technologies are capable of generating 3D point clouds and the much older radar has many other abilities. Lidar has only recently become commercially available (mid 1990s [6]) and its abilities apart from 3D point cloud generation have not been explored to nearly the extent of radar. Multispectral cameras typically record at least three frequencies in the visible spectrum (red, green, blue), and often a few more outside the visible spectrum. Hyperspectral cameras on the other hand typically have poorer spatial resolution but record hundreds of frequencies [101].

One of the most popular and widely used sensors are the Landsat Thematic Mapper and Landsat Enhanced Thematic Mapper Plus satellites [101]. These satellites have been regularly recording $30m$ GSD multispectral data of the globe since the launch of Landsat 1 in 1972 (each location captured approximately every 16 days) [101].

In comparison the use of airborne sensors for automated monitoring has encountered greater difficulty with spatial mismatching and differences in viewing angle leading to overestimations of change [38].

1.1 Urban Monitoring and Remote Sensing

Urban regions are complex with a large diversity and density of objects interconnected by highly dynamic systems. The intense demand on resources (and large economic value) forces organisations to frequently measure, analyse and comprehend these processes in order to provide services and manage for a sustainable future [99].

In the next 40 years United Nations' predicts that all of the worlds' population growth will occur in urban areas with urban populations increasing by two thirds

of their 2012 levels [96]. This growth is expected to increase the proportion of the world's population in cities from above 50% to 67% [96]. This continuing increase in urbanisation will bring even greater burdens on resources and a greater need for automated urban monitoring.

Many urban processes have a spatial element making remote sensing an ideal partner for urban monitoring. However the complexity of urban environments and organisations' desires are such that improved urban monitoring will also require other data sources and greater interdisciplinary communication [88].

Compared to general remote sensing the methods tailored to urban applications tend to use much higher resolutions, place a greater emphasis on 3D information and exploit the shape of features. These differences are driven by the high spatial complexity of cities ([49] noted that many applications require resolutions smaller than $0.5m$ GSD). At $30m$ GSD (the resolution of Landsat data) only broad landscape types such as urban, parkland and forest may be detected [99], and even with new high resolution satellites, such as the $4m$ GSD multispectral IKONOS [101], mixed pixels are a significant issue and often require sub-pixel classifiers (e.g. [70]).

Acquisition of the necessary higher resolution data is typically by airborne sensors, historically film-based stereo-photography. The inception of lidar and cheap digital photography now allow for automated remote sensing at decimetre resolutions.

At these resolutions many objects contain multiple pixels, making morphological (shape based) and object oriented analyses feasible. The more structured and repetitive features of the built environment mean shape is also much more useful than in non-urban regions.

The use of 3D information has become standard in urban applications [5] again driven by a number of factors:

- Both lidar and high resolution photography are capable of measuring depth to form point clouds (the latter through stereo matching).

- 3D data is essential for orthorectification which is required for the comparison of multiple images for monitoring, for stitching of adjacent regions and any quantitative spatial measurements (orthorectification is the process by which the image is transformed to appear as if every pixel was captured

from directly above).

- Spectrally identical land covers such as concrete roofs and concrete pavements can only be separated using 3D information.
- 3D information provides greater detail (such as heights of objects).
- Stereo views have historically been used for manual interpretation of urban scenes.

The surface created directly from the point clouds which includes all the aboveground objects is usually known as a Digital Surface Model (DSM). It can be used independently to calculate geometric features like curvature for classifications (e.g. [55]) and possibly generate virtual 3D city models. Many applications also use it in conjunction with a simulation of the ground surface as if no aboveground objects existed.

Often this surface is generated solely from the DSM or 3D point cloud (e.g. [52, 55, 85]). The name used for this bare Earth surface varies in the literature. Commonly it is referred to as the Digital Elevation Model (DEM), although some authors use DEM synonymously with DSM (e.g. [4]) or Digital Terrain Model (DTM). We use a new term, Ground Elevation Model (GEM).

The difference between the DSM and the GEM is known as the normalised DSM (nDSM=DSM-GEM). It gives the height of image pixels above the ground and is almost ubiquitous with the separation of aboveground objects from the ground.

The nDSM has been used to detect buildings [23, 71, 34], individual tree crowns [60] and assist with land cover classifications [52]. It has the potential to separate objects according to height (e.g. bushes from trees), estimate volumes, assist with habitat detection (e.g. by detecting areas with trees taller than $25m$) and generate detailed statistics (e.g. mean and variance of vegetation height).

Automated monitoring using multidate aerial data is still an unsolved challenge, despite orthorectification by high resolution DSMs. The work by Taylor et. al. [89] in manual vegetation classification and monitoring by aerial photography gives an idea of not only the magnitude and difficulty of manual interpretation, but also of the challenges that automated methods also face. Even if the same sensor has been used for all captures, differing viewing angles caused by

different flight paths create spurious change due to occlusions or deficiencies in the orthorectification. Features also manifest differently depending on the sun illumination angle and weather, both of which vary with the date and time of capture.

1.2 CSIRO's Urban Monitor Project

This research was conducted as part of the larger Urban Monitor project within Australia's Commonwealth Scientific and Industrial Research Organisation (CSIRO) to develop methods for fine-scale quantitative monitoring of urban and peri-urban environments [16]. The project aims to develop broad region automated analysis at the fine scales appropriate for application to cities including time-series of indicators of the urban built and natural environments. To this end it is partnered with urban planners and infrastructure providers, and managers of parkland, remnant vegetation, foreshore, wetland and estuarine environments. The project capitalises on the ongoing and regular collection of digital aerial photography acquired by state governments.

As a pilot study a $9600km^2$ area of interest around Perth, Australia was chosen (see figure 1.1). Digital aerial photographs of this region have been captured during the dry hot Mediterranean-like summer every year since 2007. From 2007 to 2009 the UltraCAM-D [57] was flown at $1300m$ with a field of view of 50^o capturing multispectral data (red, green, blue, near infrared) at $30cm$ GSD and panchromatic data at $10cm$ GSD. From 2010 to 2013 the sensors used were the Leica ADS40 and ADS80 which have a similar red, green, blue, near-infrared capture [15].

For all years the full dynamic range of the cameras was preserved (the digital data was not reduced to an 8 bit range, or compressed into JPEG formats). For 2007 to 2009 this resulted in approximately 13 terabytes of data per annum.

To reduce the effects of differing sensors, flight paths and atmospheric issues, algorithms for radiometric calibration [29], DSM creation and orthorectification have already been developed. Both the 2007 and 2009 datasets have been processed resulting in a 2.6TB multispectral mosaicked orthorectified image calibrated to ground reflectance and a 1.3TB DSM for each year. The GSD for each dataset was $20cm$ (the multispectral data was upsampled from $30cm$ GSD during

the orthorectification process).

1.3 Contributions

This research project developed a novel GEM generation algorithm (chapter 2) and investigated the use of Canonical Variate Analysis with Rational Polynomials (CVAR) [28] for supervised index generation and classification in urban regions (chapter 3). Previously CVAR had only been tested in a mountainous rural area using 30m resolution Landsat TM data [28].

CVAR was tested in a number of pilot regions and was able to classify multiple classes with an accuracy of 75% or more. Investigations showed that these CVAR-based classifiers required stratification for broad area application. Due to time restrictions the normalised digital vegetation index (NDVI)[78] was used to generate a two-class vegetation/non-vegetation classification for the entire region. The generation of these products and some simple derivatives over the entire region for both the 2007 and 2009 data is described in chapter 4.

With the DSM, GEM, nDSM and land cover classifications available a plethora of quantitative environmental indicators become possible.

Combining the work in this thesis with temporal-trajectories and object-oriented algorithms will help make automated monitoring from aerial photography even more powerful in the future.

1.3.1 Ground Elevation Model Generation

In order to obtain an estimate of the ground from the DSM we identified and removed aboveground objects from the DSM and fitted a surface to the remaining points. The filtering of the DSM occurred in two stages. First the DSM was segmented into regions using a slope threshold and the regions were labelled ground according to the gradient at the perimeter (algorithm from [12]). This set of ground points was then cleaned using surface fitting and a roughness threshold (this novel extension is the subject of section 2.4). Finally a high quality surface fitting was performed on the remaining ground points. The surface fitting algorithm was a multiresolution thin plate spline approach based upon [92].

The quality of the resulting GEM was investigated by a comparison to an

array of points manually interpreted as either ground or non-ground (achieved 90% accuracy), a visual comparison with a GEM created by the commercial Inpho package [62], and a manual search for large errors.

1.3.2 Classification

CVAR accepts a set of training pixels organised into groups and searches for a transformation $\frac{\mathbf{c_1}\cdot\mathbf{x}}{a+\mathbf{c_2}\cdot\mathbf{x}}$ (where $\mathbf{x}$ is the vector of spectral values of a pixel) that maximises the ratio of between-group variation to within-group variation. The constant a was fixed by the user, $\mathbf{c_1}$ was calculated from a and $\mathbf{c_2}$ using standard canonical variate analysis (CVA) [19], and $\mathbf{c_2}$ was found using Nelder and Mead's simplex search [68]. The optimal function $f(\mathbf{x}) = \frac{\mathbf{c_1}\cdot\mathbf{x}}{a+\mathbf{c_2}\cdot\mathbf{x}}$ provides the biggest separation between the training groups and can either be used as an index or be treated as a feature within a larger classification scheme.

The benefit of CVAR over usual canonical variate analysis is that the denominator enables the generation of discrimination functions resistant to the topographic illumination effect. This is the brightening/darkening of surfaces due to changes in the viewing angle and light incidence angle. It is particularly relevant to urban areas where complicated surfaces are generated by buildings, trees and other objects.

Experiments in this thesis confirmed the superiority of CVAR to CVA and developed numerous guidelines (summarised in section 3.1.6) for the use of CVAR.

The ability of CVAR to create multiclass land cover classifiers of the Urban Monitor calibrated mosaic was investigated in three pilot studies. The indices/features derived showed great ability for normalising illumination and locally separated a surprising number of cover types. However the current CVAR methods proved sensitive to variations in the spectral signature of cover types caused by differing soil types, landscape changes and limitations in calibration. Applying CVAR to the full Urban Monitor region required more stratifications than feasible within the time constraints of this research project.

For a simpler vegetation/non-vegetation classification a threshold applied to the NDVI was trialled. It proved to be less sensitive to spectral changes and reliably detected vegetation across the entire region with few parameter changes.

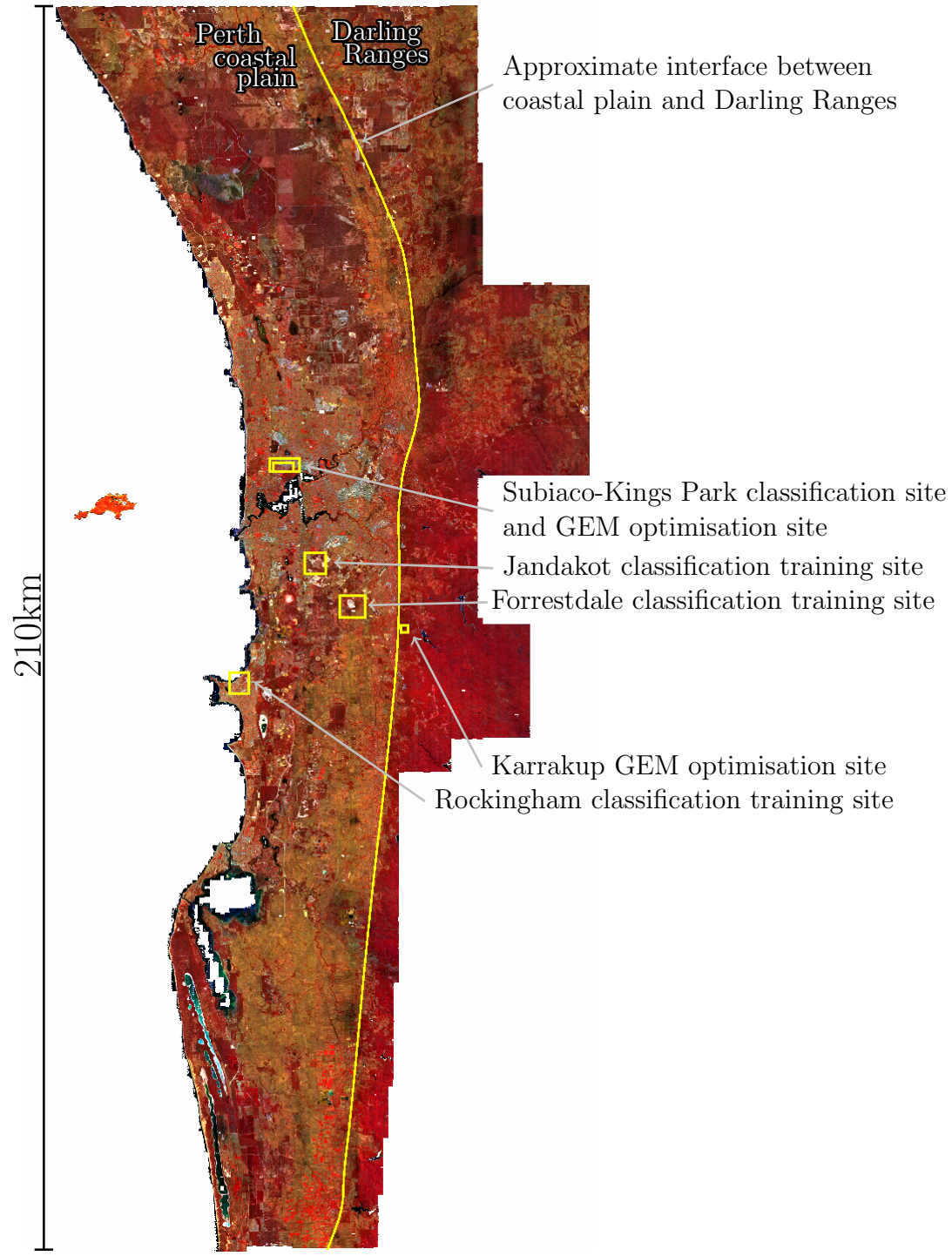


Figure 1.1: The full $9600km^2$ Urban Monitor region. Shown is the 2009 multi-spectral orthomosaick (displayed in false colour: red = near-infrared reflectance, green = red reflectance, blue = blue reflectance). Also shown are the sites used for experiments and the boundary between the two strata used for the GEM creation.

Chapter 2

Creation of Ground Elevation Models

This chapter concerns the creation of ground elevation models (GEMs) from a digital surface model (DSM) defined on a regular grid. We combine tools from spatial clustering, surface filtering and surface fitting to create a novel algorithm for generating decimetre resolution GEMs over the $9600km^2$ region.

Models of the land surface have found wide applications since the late 1950s in fields including mapping, remote sensing, land planning, engineering and geology [59].

With the ability to generate high resolution digital surface models from remote sensors the GEM has become synonymous with the filtering of aboveground features from the ground. For many applications this differentiation between bare-earth and objects is considered a necessity [84] because it enables a higher level of automated understanding [37]. Indeed our classifications (chapter 3), including the NDVI-based classification, use this height information. Whilst many algorithms exist for the automatic generation of ground elevation models there is still need for improvements to their quality [61].

Essentially any method capable of measuring depth may be used for DSM and GEM creation. The most commonly used sensor technologies are lidar, radar, or stereophotography.

Lidar directly measures the depths of points using a laser. The point clouds are usually very accurate[1] and are not effected by sun illumination (surveys can be

[1]typically better than 0.2m vertically and 1.0m horizontally for (typical) flying heights of

conducted at night). One of the biggest benefits of lidar is its ability to penetrate canopy and detect the ground beneath vegetation.

Radar is another type of active sensor that is unaffected by sun illumination but usually has poorer spatial resolution than the other sensors [54] and is also affected by viewing geometry. However it is largely unaffected by atmospheric conditions so can operate in the presence of clouds, smoke etc. The depths are derived using phase differences in the reflected signal [46].

This project used stereo-matched digital photography to generate the DSM. Stereo-matched photography suffers from feature matching errors and usually can not penetrate vegetation. However it is often collected on a regular basis and the spectral data is invaluable for wider classification schemes that involve the discrimination of more cover types than ground and non-ground.

The methods used for ground filtering can be divided into four general categories:

1. Point-to-point comparisons. e.g. the slope between two points [83], or lidar first return compared to last return [2].

2. Point-to-neighbourhood comparisons which compare points to some function of nearby points. This includes morphological transformations such as the dual rank filter [37] and comparisons to *locally* generated surfaces (e.g. the 'Robust Method' [1]).

3. Point-to-surface comparisons such as progressive densification of points labelled ground (e.g. [3] and all other TIN based methods) or progressive removal of non-ground points such as the Elastic Grid in [22].

4. Clustering/segmentation into homogeneous areas. Followed by filtration of the segments by comparison to nearby pixels or segments, or simply using geometric properties of each segment (e.g. size and perimeter). The segmentation/clustering is often performed using thresholds on slope or height, and/or morphological region growing.

The ground sampling distance (GSD) plays an important part in determining an appropriate filter as it generally becomes more difficult to distinguish objects at lower resolutions [85].

less than 1000m [6].

Most methods do not consider any spectral information. This is partly due to historical reasons, such as the scanning of photographic negatives and the prevalence of single frequency lidar, but also because methods that only require a point cloud can be inspired by other problems such as signal smoothing. The quality of current ground filters suggests that the spectral data is unnecessary. However for an example of spectral incorporation see [102] which used the DSM and spectral information to classify images and then generated a GEM from the pixels in the ground classes.

The review article [85] contains a very good discussion of the methods for ground filtering, including a comprehensive list of problematic terrains:

- *Outliers* are points that are far above or below the true surface. They are usually caused by sensor or matching errors, or birds/aircraft in the image. Due to a bias towards low points in most filters, low outliers are problematic whilst high outliers are usually handled easily.

- *Complex objects* such as very large objects, very small objects, very low objects, disconnected patches of bare earth (e.g. courtyards), or objects with complex configurations. The latter is particularly prevalent in urban environments where buildings are connected to other buildings, have multiple roof levels and are possibly overshadowed by tall trees. Disconnected patches of ground are also quite common in urban areas and their presence makes it harder to correctly filter complex building configurations.

- *Attached objects* are objects that are seamlessly connected to the ground on one edge, but are above the ground on another edge. Some examples of this are roofs that touch the ground, bridges and ramps.

- *Vegetation* particularly vegetation on slopes, low vegetation, and areas of homogeneous dense canopy[2].

- *Discontinuities and abrupt height changes* in the ground are difficult, with many filters (including the filter developed in this thesis) removing or smoothing over these terrain features.

[2]Sithole and Vosselman [85] did not include this last difficulty because their focus was lidar which can partially penetrate most vegetation

There are two types of errors in classifying ground points: omission errors (rejection of true bare-Earth points) and commission errors (false labelling of objects as bare-Earth).

After surface fitting the commission errors usually present as rough points and are consequently more obvious to humans. Omission errors in comparison are far more difficult to discern because the surface still appears smooth and realistic.

One of the most comprehensive papers comparing filters is [85] and was based upon datasets provided by the International Society of Photogrammetry and Remote Sensing. Some new algorithms still use these original datasets for comparison [67, 25] however most new methods (e.g. [54, 4, 93, 1]) have been tested on their own unique data sets and are compared to at most one older algorithm.

In section 2.5.3 our work is compared to the match-T GEM generation algorithm found in the commercial Inpho suite [62]. Match-T feature matches directly from stereo photographs. Above ground objects are considered *outliers* and removed through outlier-robust finite element surface fitting.

Sithole and Voselman [85] noted that "full automation is not possible" due to the complexity and variety of urban environments. We aimed for the errors in this ground filter to be rare and predictable, or easily noticeable by humans. As noted in [85] it is a balancing act between commissioning too many aboveground points in the surface, and omitting too many true ground points.

2.1 Algorithm Overview

In the literature it is recognised that the four different categories for ground filtering have different strengths and attempts to combine them have been made. Kobler et. al. [54] removed most non-ground objects and outliers using a point-to-point filter, and then used a novel surface based filter (a large number of surfaces are generated from random samples of the remaining points and the final ground elevation at each location is found from the distribution of the generated surfaces). Mongus and Žalik's method [67] was essentially an iterative surface fitting filter interleaved with a morphological (point-to-neighbourhood) filter. Combinations of clustering and morphological methods have also been considered [1] and examples of surface fitting with segments do exist [93]. Algorithms such as [4] and [93] may loosely be considered hybrid methods in that they cluster

the data points and use point-to-surface comparisons to filter each cluster into ground/non-ground labels. Abo Akel et. al. [1] extracted a road network using segments filtered by size and perimeter-length, and then performed multiscale point-to-neighbourhood comparisons to detect all ground points.

Our method combines a segmentation approach and a surface fitting filter. First a set of candidate ground points was generated using the segmentation filter (section 2.2). Then the commission errors in this set of points were removed using the roughness of fitted surfaces (section 2.4). Finally a surface was fitted to the cleaned set of ground points to form the GEM. The algorithms for the final surface fit and the error-cleaning surface fits were the same and are discussed in section 2.3).

The segmentation approach was ideal for our problem due to its speed, however the commission error rate was too high to solely use this method: surfaces fitted to the set of candidate ground points from the segmentation were too rough due to small commission errors on roofs and trees. Using a fitted surface to help clean this set of ground points is almost necessary[3] because a fitted surface passes height information across the gaps between the candidate ground points.

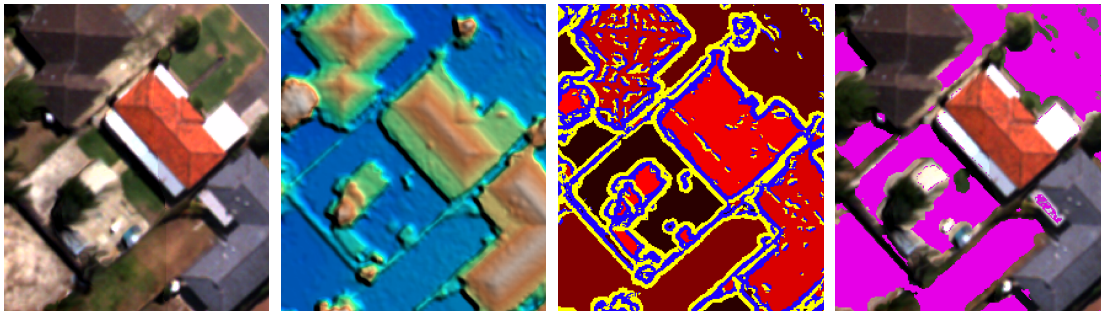
The use of surface fitting to correct a set of candidate ground points was also used by Kobler et. al. [54]. Their algorithm was designed to accept the candidate ground points from any filter, however it was only tested on a point-to-point slope-based filter.

Our surface fitting filter cleans the candidate ground points by removing points that cause large roughness in the fitted surface. Of course there are other measures by which one can determine commission errors from a fitted surface. Champion and Boldo [21] use a thin plate spline surface fitting regime and a data matching objective that decreases the influence of ground points far from the surface. Other methods such as clustering or the top-hat operator (used in [67]) are also a possibility.

[3]Multiscale neighbourhood comparisons might also work

2.2 Filtering the DSM into candidate ground points using a segmentation/morphological based algorithm

The DSM was segmented according to a slope threshold of 25^o. Segments smaller than $0.4m^2$ were automatically discarded as non-ground to avoid many errors in rough vegetation. Segments where the gradient pointed inwards for more than half of the boundary were labelled ground. All other pixels were considered non-ground. This algorithm was known as *inflows* and was developed in [12, Section 5.4.2]. An example is shown in figure 2.1.



(a) true colour (b) sun shaded DSM (c) inside inflows (d) ground mask (colour image behind)

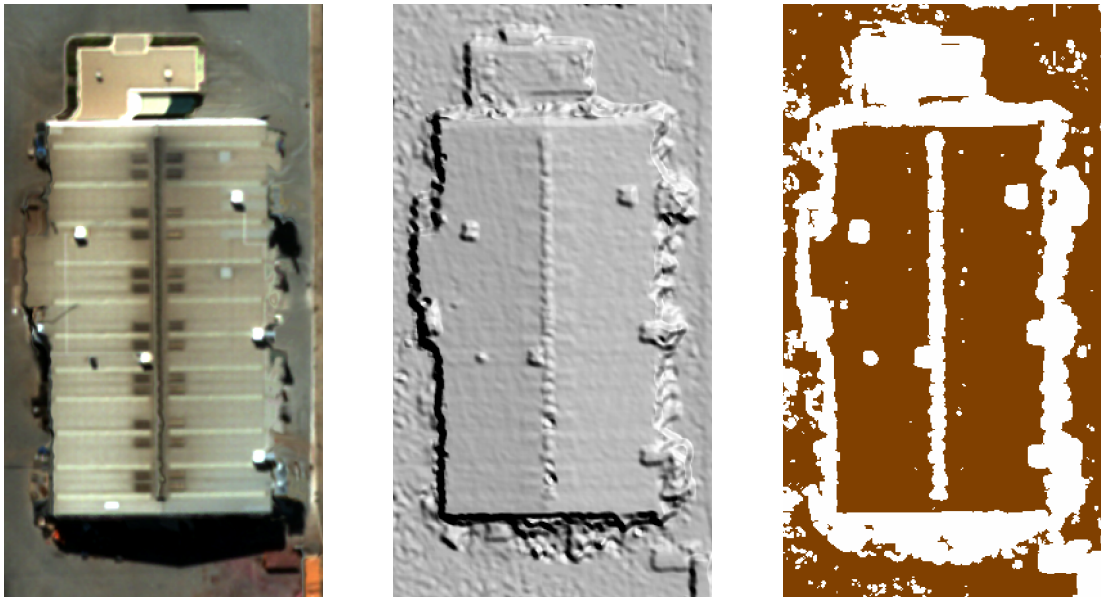
Figure 2.1: An example of the inflows segmentation filter in action. Using the DSM (b), the area was segmented into pixels above or below the slope threshold (c). In (c) pixels where the gradient flowed into the pixel are *yellow*, and pixels where the gradient flowed away from the pixel are *blue*. The number of inwards and outwards flowing pixels bounding each of the level segments was counted. In (c) the level segments are coloured with various saturations of red; the less saturated (darker) a segment the higher the proportion of inward flowing pixels. The segments with more than 50% of their boundary pixels flowing *inwards* were then labelled ground (d) (ground pixels are coloured pink).

The algorithm performed well on many of the difficulties noted by [85]: outliers in the DSM, discontinuities, sharp ridges, low vegetation, vegetation on slopes, large objects[4], and even low objects.

[4]assuming the computer loaded an area large enough to contain the entire object. In our

The slope and area thresholds were chosen to work well in nearly all locations, however the complexity, resolution and extent of the Urban Monitor region meant there were still many errors in the ground mask.

The algorithm's most troublesome objects were saw-toothed roofs, attached objects (such as roof-top car parks), roofs with many objects on top of the roof (an example is shown in figure 2.2) and roofs neighboured by even higher roofs (such as in high density commercial districts). It also had difficulty with horizontal segments in forest canopies (where the areas of segments could be greater than the area threshold of $0.4m^2$ - example shown in figure 2.3), and extremely smooth canopies such as the surface of a tall crop. It also had trouble with extremely steep terrain (greater than 25^o slopes).

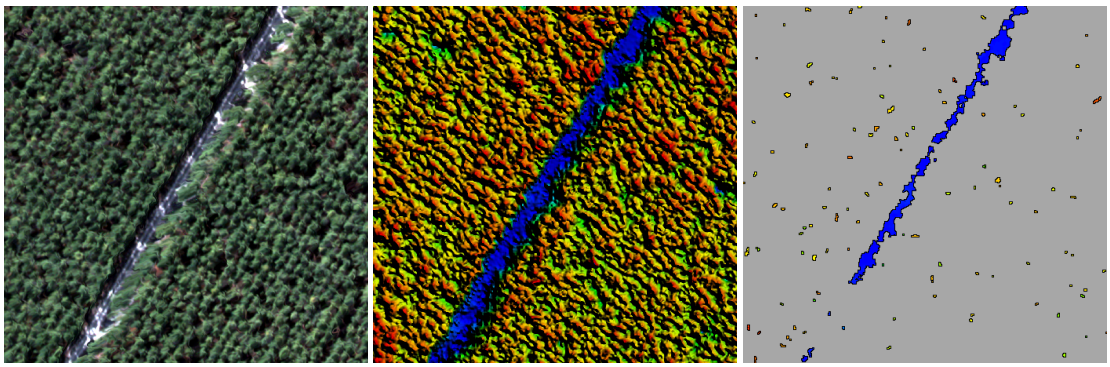


(a) true colour (b) sun shaded DSM (c) ground mask

Figure 2.2: A roof that was labelled ground due to the presence of air conditioners and a raised ridge. The air conditioners and raised ridge meant the roof segment was mostly (just over 50%) neighboured by higher pixels causing it to be labelled ground.

The presence of small area commission errors in trees, complicated roofs, and other commission errors resulted in surfaces with two to three metre spikes every few metres of horizontal distance (see figure 2.4 for an example). These errors

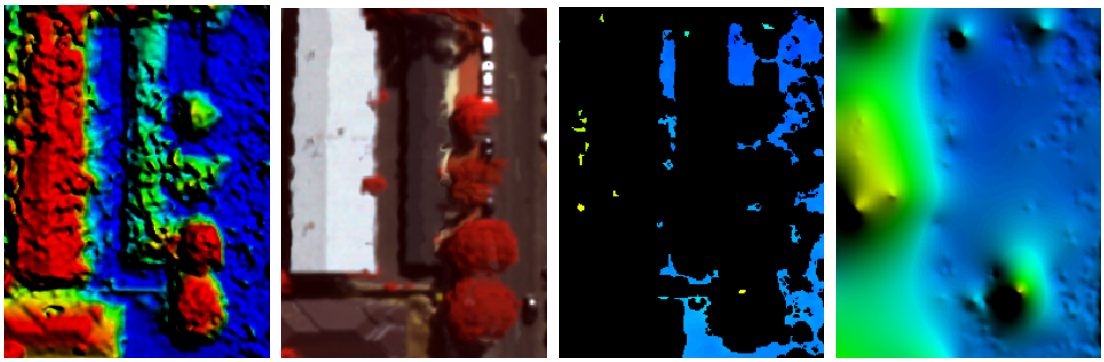
work this was $4km \times 4km$ - see chapter 4



(a) true colour (b) sun shaded DSM (c) ground mask coloured by height

Figure 2.3: A forest containing segments of canopy that inflows labelled ground. In forests canopy segments can often be labelled ground due to neighbouring taller vegetation. Here the road was correctly labelled ground, however there were many small patches at the height of the canopy that were also labelled ground. The DSM and ground mask are coloured by elevation where hot colours are high, cool colours are low. In (c) NULL pixels (pixels not labelled ground) are grey.

resulted in inaccuracies in the calculation of the heights of objects that contained commission errors and necessitated the second filtering phase.



(a) DSM (b) False Colour (c) Ground Mask (d) Fitted Surface

Figure 2.4: An example of a surface fitted to an inflows candidate ground mask. Shown is the DSM (a), a false colour display of the region (b), the input candidate ground mask (c) and the fitted surface (d). The sub-figures (a), (c) and (d) are coloured according to the same height scale (blue = low, red = high). The surfaces (a) and (d) are also sun-shaded to accentuate texture. In (c) the black pixels are those that were not candidate ground. A few high, non-ground, pixels (coloured green or yellow due to their height) can be seen in (c); they are causing the large mound-like features in (d).

2.3 Multi-resolution Surface Fitting

We fitted a thin plate spline surface to the candidate ground points using a multigrid finite element algorithm inspired by Terzopoulos [92].

Multigrid is a technique that combines computations on grids at different scales to solve boundary value elliptic partial differential equations [36]. Due to its speed and stability multigrid algorithms have been adapted to many other fields [36]. A good introductory paper on the topic is [36].

Typically the sequence of the scales visited is from coarse to fine, but not necessarily monotonic; iterative cycling through scales is often adopted. Our algorithm followed the simpler monotonic approaches of [11, 105] by starting with computations on the coarsest grid and halving the grid spacing until the desired

resolution was reached. Szeliski [87] commented that this cascadic style multigrid had the best convergence of any multigrid style tested for fitting to moderately spaced data points. Independently we found that parameter selection for the more complicated adaptive sequence in [92] was prohibitively difficult.

The multiple scales enabled both quick interpolation over sparse point clouds and fine scale fitting to dense point clouds. The thin plate spline energy optimisation problem is such that a unique, optimal surface exists in all but the most superficial cases [92]. In our implementation the data matching component dominated the thin plate spline energy and consequently the optimal surface should closely match candidate ground points in regions of dense ground points whilst be as smooth as possible in data-less regions.

In this section Terzopoulos' algorithm [92] is briefly investigated (subsection 2.3.1), our cascadic algorithm is discussed in detail and suitable parameters are established (subsection 2.3.2), and finally in section 2.3.3 there is a brief comment on support for including discontinuities in the fitting process.

2.3.1 Terzopoulos' Multigrid Algorithm

Terzopoulos' algorithm [92] fitted a smooth surface to a set of data points by minimising a thin plate spline energy $\mathcal{E} = \mathcal{E}_s + \mathcal{E}_d$ which was the sum of a regularisation penalty $\mathcal{E}_s$ and a data matching penalty $\mathcal{E}_d$. Discontinuities in the surface were enabled through a weighting function $\rho(x, y)$ in $\mathcal{E}_s$ so that the regularisation penalty for a surface $u(x, y)$ in a region Ω was defined as

$$\mathcal{E}_s(u) = \frac{1}{2} \int \int_{\Omega} \rho(x, y) \left(u_{xx}^2(x, y) + 2u_{xy}^2(x, y) + u_{yy}^2(x, y) \right) dxdy. \tag{2.1}$$

The smoothness of the surface was ignored wherever $\rho(x, y) = 0$, thus u was doubly differentiable everywhere except in $\{(x, y) : \rho(x, y) = 0\}$ where u could be discontinuous. The edge of an image was nicely modelled by setting $\rho(x, y) = 0$ on the boundary. The use of $\rho(x, y)$ in the *interior* is discussed further in section 2.3.2 however it was not in general use because a method for detecting break points had not been implemented.

The matching penalty that Terzopoulos' used was

$$\mathcal{E}_d(u) = \frac{1}{2} \sum_{(x_i, y_i) \in \mathcal{D}} \alpha_i (d_i - u(x_i, y_i))^2 \tag{2.2}$$

where $\mathcal{D}$ was the set of data points (x_i, y_i) with heights d_i that the surface was desired to fit to. Often these data points are known as *control points* or *depth constraints*. In our case they were the set of candidate ground points.

Terzopoulos' algorithm [92] discretised the surface into an array of square elements. Within each element the surface was thought of as a polynomial defined according to the heights of the corners (known as *nodes*) of the elements.

In this discrete setting the penalty functions were defined to approximate the continuous versions

$$\mathcal{E}_s(u) = \frac{1}{2A} \times \sum_{(i,j)} \rho(i,j) \left(u_{xx}^2(i,j) + 2u_{xy}^2(i,j) + u_{yy}^2(i,j) \right)$$
$$\mathcal{E}_d(u) = \sum_{(i,j)} \alpha(i,j) \left(u(i,j) - d(i,j) \right)^2 ,$$

where A was the area of an element, (i,j) denotes the nodes, $\alpha(i,j) \neq 0$ only if the node (i,j) was a control point, and $d(i,j)$ was the height of the control point. The derivatives u_{xx}, u_{xy} and u_{yy} were approximated by comparisons with neighbouring nodes.

We will sometimes be considering the penalties associated with an individual node of a surface $u(i,j)$

$$\mathcal{E}_s(i,j) = \frac{1}{2A} \rho(i,j) \left(u_{xx}^2(i,j) + 2u_{xy}^2(i,j) + u_{yy}^2(i,j) \right)$$
$$\mathcal{E}_d(i,j) = \alpha(i,j) \left(u(i,j) - d(i,j) \right)^2 .$$

When (i,j) was not a control point then $\alpha(i,j) = 0$ and the optimal height of $u(i,j)$ is the smoothest surface between the heights of nearby nodes.

If $\rho(i,j)$ is known and there are three non-collinear control points for each connected region, then there exists a function unique up to the boundary that minimises $\mathcal{E}$. The proof of this is very similar to the smooth version (see [91, Ch.5]) although some details are included in appendix A.

To approximate the solution the algorithm started with an initial guess $u : \mathbb{N}^2 \to \mathbb{R}$ and iterated through each node (i,j) modifying the surface $u(i,j)$ to make[5]

$$\frac{\partial \mathcal{E}}{\partial(u(i,j))} = \frac{\partial \mathcal{E}_s(u)}{\partial(u(i,j))} + \frac{\partial \mathcal{E}_d(u)}{\partial(u(i,j))} = 0.$$

[5]The details of this equation can be found in appendix section C.2.1 or [92, eq.20-21]

Each pass of the image is called a *relaxation*, the exact form used in our implementation is called *Gauss-Seidel Relaxation* (see [91, B.1]).

In most cases relaxations are repeated numerous times before an acceptable surface is reached making it by far the most time consuming step. Multigrid algorithms significantly reduce the number of relaxations required by using relaxations at coarser resolutions to correct the fine resolution surface. A convenient method to describe the computational cost of a multigrid algorithm is then in terms of a *Work Unit*, where a Work Unit is the number of computer operations required for a single relaxation at the finest resolution.

The set of equivalent surface fitting problems at different resolutions is often thought of as a *pyramid* with the first level (called *level* 0) the coarsest. In our case each coarser level corresponded to a doubling of spacing between nodes. In multigrid algorithms the coarse level problems typically do not fit directly to the control points, but solve for the error of the finer levels [36]. In other words at coarser levels the problem is reinterpreted such that the coarse level solved for the *difference* between the fine surface and the ideal solution. Thus only at the finest level did relaxations solve for a surface u such that $\partial\mathcal{E}/\partial(u(i,j)) = 0$.

Terzopoulos' algorithm transferred between finer and coarser levels adaptively. If the relaxations were proceeding slowly then computations were moved to a coarser level, if the surface had achieved some finish criterion then computations were transferred to a finer level.

Results and Analysis of Terzopoulos' Algorithm

After implementing the algorithm it was found that choosing good parameters was prohibitively difficult and that directly fitting to depth constraints at coarse levels performed better than solving for the residual of fine levels. Investigations on pyramids with a number of levels and a number of different data sets occurred.

When left to operate adaptively the algorithm often oscillated between levels for a very long time, irrespective of whether the ground elevation or the residual was being fitted. The issue was that the speed criterion would cause the algorithm to transfer to the next coarsest level, even when coarser relaxations could not improve the surface. Choosing speed and convergence parameters that prevented many oscillations proved very difficult.

In the following a thorough comparison between pyramids with only two levels

is described. The pyramids were used to fit a surface to an image containing both dense and sparse candidate ground points (figure 2.5a). At the coarse level fitting was conducted on either the coarsened control points, or the residual of the fine resolution surface. The algorithm was forced to oscillate between level 0 and level 1 ten times in both cases. The results were compared to when the algorithm was forced to enter level 0 (the coarse level) once.

The comparisons with high numbers of oscillations were conducted because in initial investigations the algorithm oscillated many times between levels. Ideally parameters would be found that lead to only a few oscillations, hence the tests where no oscillations occurred. The final surfaces (figure 2.5) show that solving for $\partial\mathcal{E}/\partial(u(i,j)) = 0$ at the coarse level performed significantly better, regardless of the number of oscillations. It was better at filling in both small and large data-less areas, and yielded much lower final energies in both cases.

2.3.2 The Cascadic Algorithm

Following the ideas of [105] Terzopoulos' algorithm was simplified into a *cascadic* multigrid which started at the coarsest scale and methodically worked down to the finest scale, never returning to the coarser levels. Each level in the pyramid fitted a surface to coarsened depth constraints and had a fixed number of relaxations which were chosen by the user for each application.

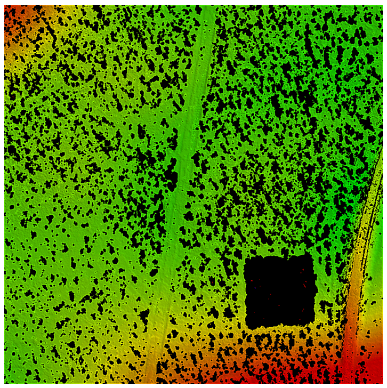
These changes may not have led to an algorithm with optimal complexity but they made the behaviour far more reliable and predictable which was especially valuable for processing large regions automatically.

Experimenting with this algorithm also provided valuable understanding of multigrid surface construction should the need for a more complicated multigrid strategy ever arise.

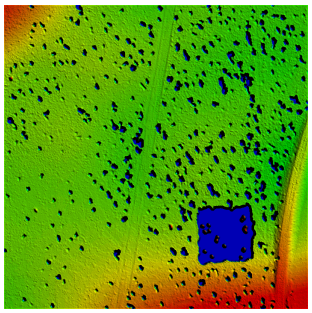
The control points weights $\alpha(i,j)$ were set to $\alpha(i,j) = 1/A$ to allow $1/A$ to be factored out of the relaxation equation. Without this change floating point errors caused relaxations at the coarsest levels to prematurely converge.

The initial surface guess (which was at the coarsest level) was the coarsened height data. Originally the initial surface guess was 0 where no height data existed but this was later changed to be the average control point height.

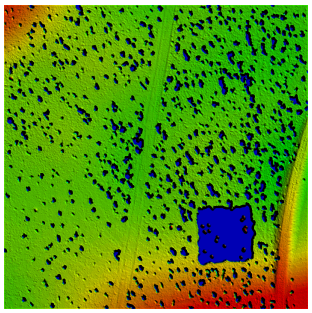
More information on the finite elements used, the relaxation method (Symmet-



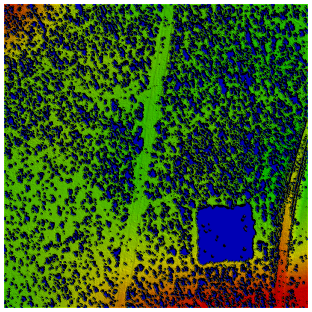
(a) Input Candidate Ground



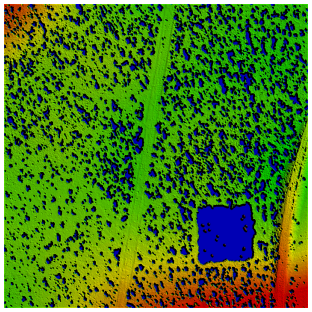
(b) few oscillations



(c) many oscillations



(d) few oscillations; solved for residual



(e) many oscillations; solved for residual

Figure 2.5: Surfaces generated by Terzopoulos' algorithm where the coarse level relaxations fit a surface to coarsened versions of the control points (b and c) or where the coarse level solved for the residual (d and e). The input set of candidate ground points is shown in (a). Cool colours correspond to low elevations, hot colours are high elevations, black is NULL. Pyramids with two levels were used in all cases. For (b) and (d) the algorithm started in level 1 (the finest level), transferred to level 0 (the coarsest level), and then moved back to level 1 before finishing. For (c) and (e) the algorithm began with level 1 and then oscillated between level 0 and 1 ten times before finishing in level 1. Observe that many more NULL patches have been smoothed over in the top row, suggesting that fitting to coarsened control points is much better than fitting to the residual, irrespective of the number of oscillations.

ric Gauss-Seidel), interlevel transfers, implementation in C++ and many other details can be found in appendix C.

In the remainder of this section the optimal number of levels is found experimentally, the behaviour of relaxations discussed and the best number of relaxations for each level investigated. The parameters used in the final algorithm are then summarised.

Number of Levels Heuristically we were looking for the surface that best matches the data and is smooth where no data exists. Thus there was little to be gained from multiple levels fully populated by control points because in these levels any surface fitting *smooths* the data. However the coarsest level should be close to fully populated otherwise many relaxations are required to produce a smooth interpolation across data-less regions.

This assessment was tested empirically by fitting to a range of candidate ground point sets, from sparse to dense, with varying numbers of levels in the pyramids. See appendix B.1.1 for more details.

Although more tests are required to reach a definitive conclusion, the results suggest that the best surfaces are produced when all nodes in the coarsest level contain a depth constraint.

For the Urban Monitor region the coarsest level grid spacing was consequently set to $102.4m = 0.2m \times 2^9$ which corresponded to a ten level pyramid. This was because regions of $102.4m \times 102.4m$ or larger that contained absolutely no visible bare earth were only likely occur over water, large industrial warehouses or dense forests.

Change caused by a Gauss-Seidel relaxation Consider the behaviour of the surface in three different situations:

1. Areas of dense control points. The surface inherited from a coarse resolution fitting is likely to be too smooth in areas of dense control points. This is because the coarse level surface is fitted using fewer grid points (greater spacing between grid points) and then transferred using smooth polynomial interpolation. Because of the emphasis on data matching, after a transfer the surface changes quickly to match the new variations in the height data. The pixels adjacent to data areas then undergo moderate change in order to

minimise the smoothness energy. At each successive relaxation this effect is passed on to pixels further away from the control points.

2. Small patches of outlying points some distance from other control points (these are the small inflows commission errors). It is difficult to predict the behaviour of relaxations in these areas. On the one hand the distance (in nodes) to good control points doubles with each level so the surface can accommodate larger height changes and thus get closer to outliers. However the data matching weight $\alpha(i, j)$ reduces by a factor of 4 with each level so smoothness also becomes more important.

3. Points far away from any control points. From multigrid literature (e.g. [36]) relaxations like Gauss-Seidel quickly smooth across short distances but are very slow over large distances. Behaviour of the initial relaxations thus depends on the smoothness of the transfer from the coarse level. Our transfer is a smooth polynomial interpolation (see section C.2.1) so the initial changes are likely to be very small. The effect of disturbances near control points will slowly transfer through the nodes so after some relaxations larger changes may occur.

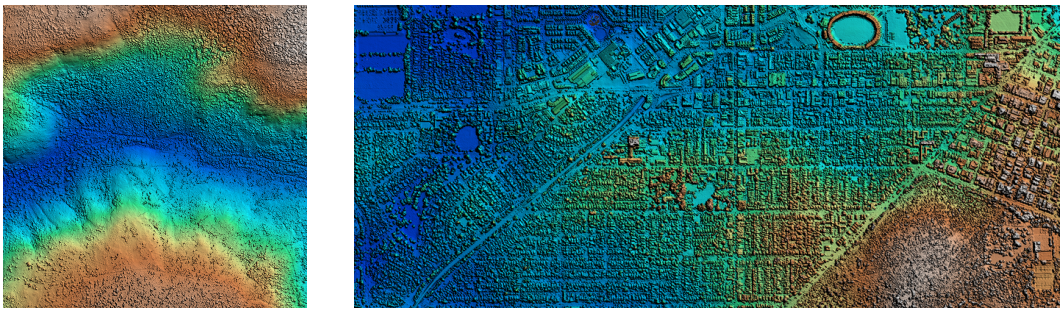
Logs showing $\mathcal{E}_s$ and $\mathcal{E}_d$ after each relaxation, and snapshots of the energy distributions $\mathcal{E}_s(i, j)$, $\mathcal{E}_d(i, j)$ for the surface fits in appendix B.1 were used to confirm these predictions.

Nodes in isolated patches of maxima/minima increased in roughness and despite the smaller emphasis on data matching the surface moved closer to these patches at each relaxation. See appendix B.1.2 for more details.

The Number of Relaxations Simple Monte Carlo searches were used to determine broadly usable relaxation amounts for the Urban Monitor data. The energy $\mathcal{E}$ of the final surface was used to rank each test. Although $\mathcal{E}$ had shortcomings (the smoothness of data-less areas was almost negligible compared to the energies near control points) time constraints prevented development of a more suitable measure.

The searches were conducted on the inflows-generated candidate ground mask of two test regions: an area in Karrakup (see figure 2.6a) containing a large amount of vegetation and very steep valley walls; and an area of Subiaco-Kings

Park (see figure 2.6b) containing large trees, native vegetation, commercial buildings, many residential houses, railway tracks and a railway tunnel.



(a) Karrakup (b) Subiaco-Kings Park

Figure 2.6: The two regions used to experiment with relaxation amounts.

The pyramids for all tests contained 10 levels with the coarsest level at a resolution of $102.4m$. The number of relaxations for the coarsest 7 levels were chosen randomly between 0 and 1000. Due to computational limitations no relaxations were performed in the finest three levels. Any surface fits that would cost more than 0.60 work units or 2.0 work units for Karrakup and Subiaco-Kings Park respectively were skipped. 1100 different relaxation combinations were tested in the Karrakup region and 1136 different combinations were tested in the Subiaco-Kings Park region.

A summary of the best forty surface fits for each region is shown in figure 2.7. Also shown is the distribution of the relaxations of the 900 surface fits with poorest energies. The distribution of the top forty surface fits was considered because this experiment aimed to discover relaxation amounts that worked well in both regions, rather than finding the absolute best surface fit for each region. The size of these groups, 40 and 900, were chosen to include the very best and everything significantly poorer than the very best respectively. In hindsight if these groups were chosen more empirically, such as searching for clusters in the final energies, then more precise trends may have surfaced.

The distribution of relaxation amounts at the $102.4m$ resolution (level zero) was close to uniform for both regions which suggested that the number of relaxations did not greatly affect the final energy. This is not surprising given that there were very few non-control points at this resolution and the relaxations were extremely cheap (4^6 times cheaper than the level six relaxations). A similar

conclusion may be made about levels one and two, although there appears to be some tendency for high numbers in level one. These three levels were set to $200, 400, 300$ for draft fits or $400, 500, 400$ for final high quality fits. The high numbers were to remove as much dependence on the initial guess as possible, and to ensure any (rare) large data-less regions were smoothed over.

In Karrakup a significant decrease in relaxations occurred between levels two, three and four. In Subiaco-Kings Park there was no significant change before level three, but a large decrease did occur between level three and level four. In the finer levels (levels four, five and six) the relaxation amounts stayed fairly constant in both regions. Typically relaxation amounts that *did not* produce good surfaces (in red in figure 2.7) also *did not* include these sharp drops and the plateaux of relaxation amounts at the fine levels, suggesting that it was important for finer levels to maintain a reasonable amount of relaxations. The cause for this phenomena was unknown and it was also unknown how these numbers would be effected if relaxations were performed in level seven and finer.

In the absence of more informative experiments and analysis the relaxations at resolutions of level three and finer were usually governed by a piecewise linear function with points given at level two (300 or 400 relaxations), at the finest level and a point halfway between level two and the finest level.

Final Cascadic Algorithm

The final algorithm used for cascadic surface fitting contained a ten level pyramid with the resolution halving at each level. For problems where greater speed and smoother surfaces were required the finer resolutions were ignored (this occurred during surface based filtering in section 2.4).

The relaxations for lower quality surface fitting were determined using three parameters a, r_1 and r_2:

1. Levels 0, 1 and 2 always contained 200, 400 and 300 relaxations respectively.

2. The relaxations in the remaining levels were determined by a piecewise linear function through the points $(2, 300)$, $(2 + \frac{a-2}{2}, r_1)$, and (a, r_2). Where a, r_1 and r_2 were chosen by the user. a was the finest level in use (not necessarily level 9) and r_1, r_2 were the relaxations at the midpoint and the finest level respectively.

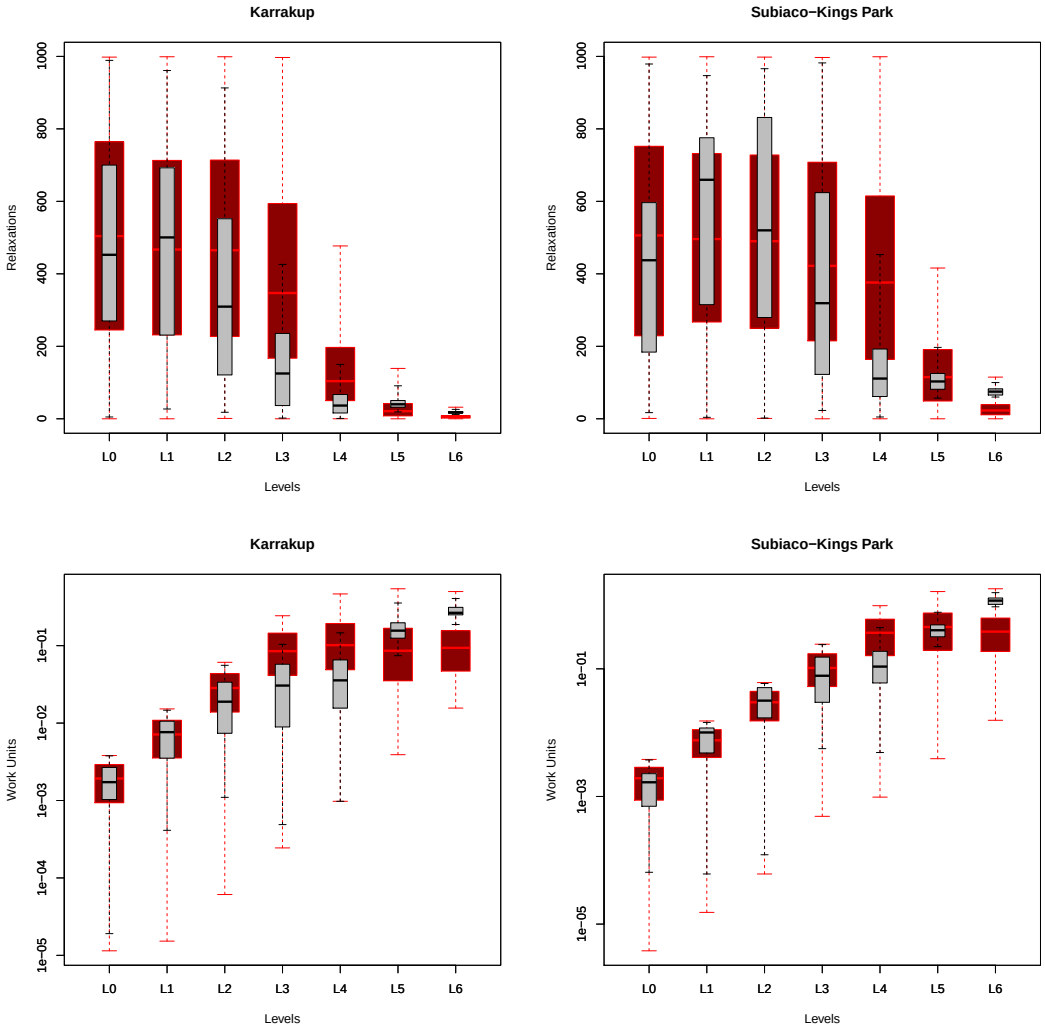


Figure 2.7: The relaxations used for the best 40 surfaces in the Karrakup and Subiaco-Kings Park test regions. Box and whisker plots of the distribution of the relaxations *(top)* and work units *(bottom)* are shown. In the backgrounds (in red) is the distribution of 900 surface fits that were not very good. These plots show that it was important for the amount of work units to continue increasing with each level, or equivalently it was important for there to be a high number of relaxations even at the finest resolution.

The relaxations amounts for the final, high quality surface fitting were

L0	L1	L2	L3	L4	L5	L6	L7	L8	L9
400	500	400	350	300	250	150	100	80	20

which corresponded to 49.9823 work units. This was not quite the piecewise linear pattern used for low quality surface fitting: the amounts in levels six, seven and eight have been increased for those occasional locations where the terrain is extremely rough. The computational cost precluded the finest levels from being increased as well.

Various minor improvements to the algorithm occurred throughout the experimentation, the most significant being:

- Coarse nodes were centred between, and equidistant from, four fine level nodes. This resulted in large improvements on steep terrain containing little data. Previously the coarse nodes were collocated with every second fine level node.

- The discontinuity parameter $\rho(i,j)$ in the regularisation energy was split into three parameters representing a discontinuity in the x direction $\rho_x(i,j)$, a discontinuity in the y direction $\rho_y(i,j)$, and a discontinuity in both directions $\rho_{xy}(i,j)$ (see section 2.3.3 for more detail). Thus the regularisation energy was

$$\mathcal{E}_s(u) = \frac{1}{2A}\sum_{i,j} \rho_x(i,j)u_{xx}^2(i,j) + 2\rho_{xy}(i,j)u_{xy}^2(i,j) + \rho_y(i,j)u_{yy}^2(i,j).$$

  This yielded much smoother surfaces on the boundaries of the image because the smoothness parallel to the boundary was factored into the relaxations.

- Every second Gauss-Seidel relaxation started at the bottom right node of the image, so two relaxations were equivalent to a symmetric Gauss-Seidel relaxation.

More details on the implementation can be found in appendix C.

RAM Requirements. The algorithm required the complete input data pyramid and surface approximations to be kept in memory so the RAM required was

at most

$$\left(2+\frac{1}{4}\right)\frac{4}{3}d\left(1-\frac{1}{4^{L-1}}\right)+200L\text{bytes}$$
$$=3d\left(1-\frac{1}{4^{L-1}}\right)+200L\text{bytes}.$$

Where d was the memory required to store the input ground mask and L was the number of levels in the pyramid. If discontinuities were in use a further $\frac{1}{3}d\left(1-\frac{1}{4^{L-1}}\right)$ bytes of memory were required to store the pyramid of discontinuity maps.

2.3.3 Surface Fitting with Discontinuities

Terzopoulos' solution to discontinuities was to inhibit certain parts of $\frac{\partial\mathcal{E}}{\partial(u(i,j))}=0$ in the relaxation calculations [92]. These inhibitions could not be generated by setting $\rho(i,j)=0$ which meant that the relaxations were no longer minimising the energy $\mathcal{E}$, and existence and uniqueness may no longer be guaranteed. If the treatment of discontinuities originated from the energy function then it would be easier to discern the requirements for a unique solution especially if we could use the $V-$*ellipticity* test (see appendix A).

In the original thin plate spline energy the only option was to set $\rho(i,j)=0$ which removed more smoothness than necessary. For example, suppose the pixels (a,b) and $(a+1,b)$ were separated by a discontinuity. Then any $u_{xx}(i,j)$ that compared them should be ignored and with the original energy function that meant $\rho(a,b)=\rho(a+1,b)=0$. But then $\mathcal{E}_s$ also ignored $u_{xy}(a,b)$ and $u_{yy}(a,b)$. A better option is to divide ρ into three different functions, one for each of u_{xx}, u_{xy}, and u_{yy}. Thus the *new* smooth energy potential was

$$\mathcal{E}_s(i,j)=\frac{\rho_a(i,j)u_{xx}^2(i,j)+2\rho_b(i,j)u_{xy}^2(i,j)+\rho_c(i,j)u_{yy}^2(i,j)}{2A}.$$

This resulted in surfaces with greater smoothness *parallel* to discontinuities.

It is easy to see that the existence and uniqueness proof (appendix A) also applies to this energy function. In fact the requirements for uniqueness are slightly weaker with only some corners of discontinuities requiring non-zero $\alpha(i,j)$.

The relaxation function needed access to the discontinuity functions $\rho_a(i,j)$, $\rho_b(i,j)$, $\rho_c(i,j)$ to determine the appropriate form of $\frac{\partial\mathcal{E}}{\partial(u(i,j))}$ for each pixel. In our

implementation the discontinuity functions were stored in an 8 bit image pyramid. Unfortunately this required additional memory and the extra calculations caused a significant reduction in speed.

2.4 Outlier Detection using Surface Fitting

There were two types of errors present in the candidate ground points: (a) non-ground points that the morphological filter failed to remove and (b) errors in the DSM itself, with the former being much more frequent and influential.

Despite these errors forming a relatively small proportion of the data, they sometimes adversely affected large areas. For example high control points some distance from other control points pulled the surface upwards to create hills, or entire roofs that (erroneously) remained in the ground masked DSM created large two metre high features in the landscape irrespective of the proximity to true ground points (see figure 2.4 for an example).

The literature contains a number of methods appropriate for removing these erroneous control points. Some have used the absolute difference of control points to a fitted surface ([21], [42]), whilst [54] generated surfaces using random samples of the control points and compared all the resulting surfaces. In general any filter designed to remove non-ground objects using surface fitting was likely to succeed because the proportion of errors in the candidate ground mask was very low. The use of surface fitting was important because it transmitted information across the gaps in the candidate ground mask; it meant that the outliers that were some distances away from other points could be removed using simple local operations.

Whilst investigating multigrid surface fitting it was observed that with each relaxation the surface increased in roughness near isolated minima and maxima (section 2.3.2). This was the core idea behind our outlier detection strategy. It was particularly easy to implement because it used measures that were natural to the thin plate spline regime.

To remove large patches of error (such as entire roofs) outlier removal was repeated six times. Each time the following was performed:

1. A multigrid surface fit of the current set of control points.

2. Calculation of $\mathcal{E}_s(i,j)$ at the location of each control point (i,j).

3. Removal of control points according to a threshold on $\mathcal{E}_s$ and whether the control point was above or below the fitted surface.

The simplicity of this method made it flexible and easy to control.

In the next section experiments are presented that establish good combinations of relaxation amounts, roughness thresholds and when to protect control points below the fitted surface. The final parameters were such that all six outlier detection phases required less than half the computations of the high quality surface fit given in section 2.3.2.

The differentiation between control points above or below the surface reflected the fact that the majority of errors in the masked DSM were due to commission errors. The bias towards low points is a fairly common practise in GEM creation especially for lidar data (e.g. [54]). If this bias was not present then the roughness threshold would have removed approximately the same amount of true ground points as erroneous points. This was because the outlying control points created roughness at the nearest true ground points as well.

For outlier removal care was taken to avoid fitting the data too well. This was because a surface that fits well to any (erroneous) patches of large planar roofs will be very rough on the outermost edge of the roof, and smooth on the interior; whilst a smoother surface fit will result in the roughness extending further into the roof. Consequently the surface fitting was stopped at a coarse resolution before reaching the finest level. An important benefit of these reductions in relaxations was that the time required for surface fitting was much smaller than needed for the high quality surface fitting.

2.4.1 The Parameters Used for Outlier Detection

The method for choosing parameters involved a combination of heuristics with a small amount of experiments. The dependence on heuristics was necessary because no test region or numerical measure sufficiently captured all the variations of the Urban Monitor region.

The area can roughly be stratified into a flat sandy coastal plain and the hilly, forested Darling Ranges (see figure 1.1). Exploratory testing of error removal gave a strong indication that different parameters would be required for each terrain type. In the Darling Ranges the valleys and ridges were much steeper and sharper,

and the ground was obscured by more trees, so the roughness thresholds needed to be higher. On the smoother, flatter, plains stricter thresholds could be used to remove more errors.

For each stage of error removal the resolution, numbers of relaxations, an $\mathcal{E}_s$ threshold, and whether to remove lower-than-surface outliers was decided. Given that all these choices had to be made with a combination of heuristics and experiments it was decided that six outlier removal stages was the limit of complexity.

Guiding principles in choosing the parameters were:

- With better surface fitting and higher resolutions the *extreme* roughness caused by outliers becomes more concentrated around the edges of data, especially near errors in the data. Thus more relaxations improved the detection of the edges of patches of outlying data. However more relaxations also meant the roughness was more concentrated and less of the interior of outlying patches was removed.

- Removing a low resolution data point removed larger amounts of data than fine resolution pixels. This was because the removal of a low resolution pixel was achieved by removing all the fine resolution pixels that it covered.

Parameter Selection for the Darling Ranges

An area in Karrakup containing a particularly large, extremely steep valley (see figure 2.8 and figure 1.1 for the location) was chosen to test parameters for outlier removal. This test region contained no roofs which was tolerable because the segmentation/morphological filter only had trouble with certain types of roofs (see section 2.2) of which there were very few in the Darling Ranges.

The first error removal stage was chosen to occur at a resolution of $3.2m$. At this resolution small patches of error that were within $3.2m$ of true ground points were usually smoothed out, but all other errors remained. If roofs had existed in the Karrakup ground mask subset then their edges would not have been overly smoothed. Unfortunately at this resolution patches of isolated ground (such as back yards or road segments) would have been just one or two pixels and any nearby errors could have caused the surface to be rough. To avoid removing these patches of ground the first outlier stage was set to only remove control

points above the surface. Surface fitting to low resolutions was very quick so this first stage provided a quick way to remove the large and isolated errors.

The next three stages of removal occurred at $1.6m$ resolution. At this resolution patches of road (at the bottom of the valley in Karrakup) and errors in trees can be discerned (and if there were any roofs they would also be found). Heuristically this is an ideal resolution for cleaning the masked DSM in a hilly landscape: finer resolutions are far more costly, and remove fewer control points, whilst coarser resolutions smooth out valuable patches of ground data.

However it was still necessary to clean up the very small errors, so the final two stages moved to finer resolutions of $0.8m$ GSD and $0.4m$ GSD. Outlier detection at the full $0.2m$ resolution was not considered due to the greater time it required.

Initial guesses of the smooth energy thresholds were very stringent with an aim to find the strictest possible thresholds for Karrakup. These parameters should produce good results on any shallower valleys whilst removing the maximum possible number of errors. Initially there were three stages where rough control points below the surface were protected from removal. The relaxation amounts were chosen according to the trends found in section 2.3.2.

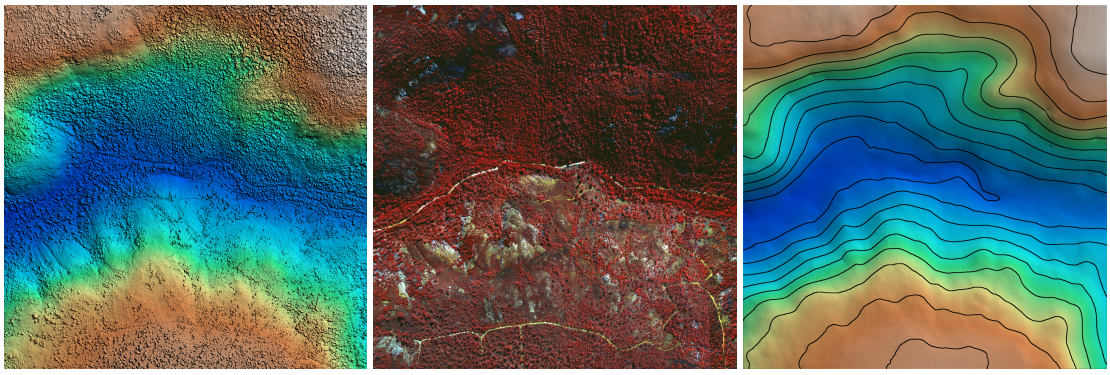
These original parameters led to a ground elevation that missed the shape of the ridges and a valley that had been significantly smoothed. Almost no data from the valley floor or walls remained after the outlier removal. After several tests where relaxations were increased and the energy thresholds loosened, the biggest improvement occurred by increasing the number of stages in which below-surface control points were protected. This search for better parameters and the results is described in more detail in appendix B.2.1.

The final parameters (table 2.1) generated a ground elevation (see figure 2.8) that was of sufficient quality considering that Karrakup was close to the steepest valley in the Urban Monitor region. The valley floor and walls were well modelled, with only some texture lost. The majority of the inaccuracies were due to a omissions in the ground mask, which could not be overcome by outlier removal. It was likely that these parameters produced GEMs of equal or better quality in other hilly areas.

Using these parameters the GEMs produced in Darlington, a hilly area that contained houses, and Mt. Helena, another hilly area, were of similar quality to Karrakup; the Mt. Helena region contained only two noticeable errors.

Parameters	Stages					
	1	2	3	4	5	6
rmv. below?	n	n	n	n	y	y
$\mathcal{E}_s$ thresh.	250	70	70	30	30	20
lvl 9 relax						
lvl 8 relax						40
lvl 7 relax					50	60
lvl 6 relax		80	70	60	75	80
lvl 5 relax	90	90	85	80	100	100
lvl 4 relax	100	100	100	100	125	166
lvl 3 relax	110	200	200	200	225	232
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Table 2.1: Outlier detection parameters for the Darling Ranges. This represented 25.8178 work units, half the cost of the high quality surface fitting. Surface fitting to the candidate ground points after this process yielded a GEM that closely matched the terrain.



(a) sun-shaded DSM (b) false colour display (c) final GEM with contours

Figure 2.8: The region of Karrakup used to guide parameter choices in the Darling Ranges. Grey colours correspond to high elevations, cool colours correspond to low elevations. The final GEM captured the shape of the valley with only slight smoothing of the North-West ridge, the loss of a small knoll on the valley wall and some smoothing of very steep slopes. See figure B.4b for more detail.

Parameter Selection for the Coastal Plain

On the coastal plain the built environment dominated and there were many more complicated aboveground structures than in the hills. It was desirable to clean more of these errors by taking advantage of the greater smoothness of the landscape.

An area covering Kings Park and Subiaco in central Perth was used to trial parameters (figure 2.9; see figure 1.1 for location). This area contained some natural bushland, landscaped parks, complicated commercial roofs and residential houses with large overhanging trees. The ground mask in this area was probably the most complicated and error prone of any urban area that these parameters were designed for. It was decided that these parameters would not be designed for industrial/commercial centres due to the large amount of inflows commission errors in these regions (these errors needed to be manually corrected after the GEM was created - see section 4.1.3)

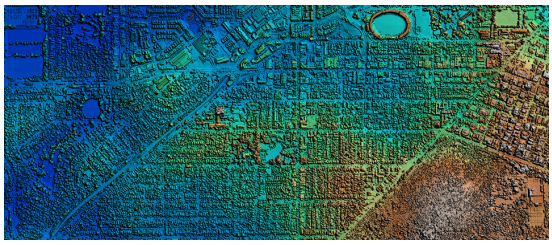
The final parameters for the Darling Ranges provided the starting point for testing parameters for the coastal plain. With these parameters the final surface captured most of the landscape, preserved all the roads and removed many small commission errors, however many large roof commission errors still remained (see figure B.6a). The experiments for determining better parameters for the coastal plain are described in detail in appendix B.2.2.

The final outlier detection parameters (see figure 2.9c for the GEM and table 2.2 for the parameters) removed all but three roof errors, but also removed part of the train tracks. Removal of the remaining errors would have resulted in further degradation of the train tracks.

2.5 Results of the Ground Elevation Model Algorithm

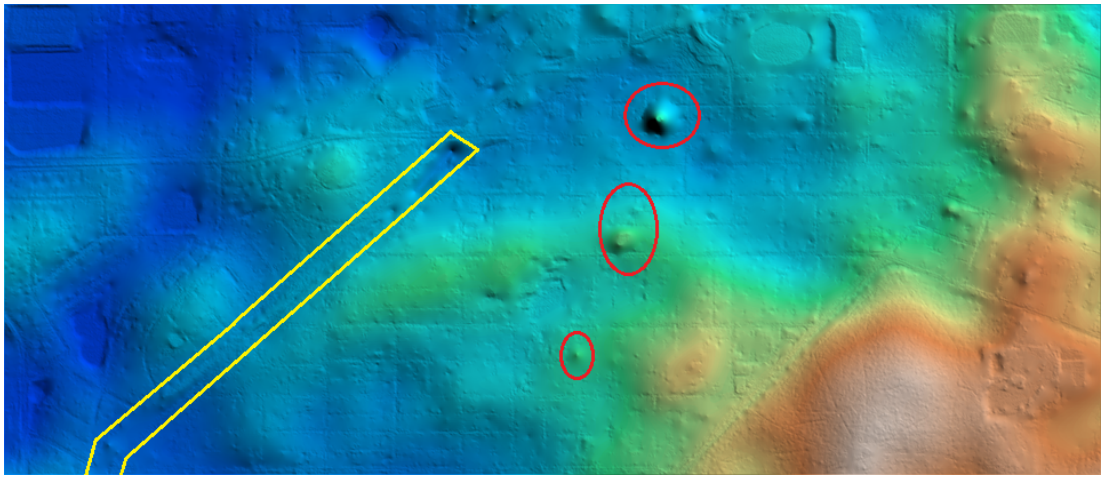
Chapter 4 discusses the application of the GEM algorithm to the full Urban Monitor region for two dates.

It was not possible to construct the entire ground elevation of Perth in one piece. Instead GEMs were created for windows of 20000 by 20000 pixels ($4km$ by $4km$) with an overlap of 1000 pixels ($200m$) and the results feathered by linear



(a) sun-shaded DSM

(b) false colour display



(c) final sun-shaded GEM

Figure 2.9: The Subiaco-Kings Park region used to guide parameter choices for the Coastal Plains. Grey colours correspond to high elevations, cool colours correspond to low elevations. Three major roof errors still remained after the outlier removal (red ellipses) causing hill-like features in the GEM, however further roughness removal was likely to remove more terrain in the railway reserve (yellow polygon).

Parameters	Stages					
	1	2	3	4	5	6
rmv. below?	n	y	n	n	y	y
$\mathcal{E}_s$ thresh.	100	40	30	30	30	20
lvl 9 relax						
lvl 8 relax						40
lvl 7 relax					50	60
lvl 6 relax				60	75	80
lvl 5 relax	90	90	70	80	100	100
lvl 4 relax	100	100	100	100	125	166
lvl 3 relax	110	110	130	200	225	232
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Table 2.2: The final outlier detection parameters for the Perth coastal plain. The total work unit cost was 23.3764.

weights that increased from 0 on the edge of a window to 1 where the overlap finished. The GEM of the entire area was then manually inspected for large errors or large patches of errors.

The GEM was also compared in detail to the commercially available match-T algorithm (section 2.5.3). The accuracy of the GEM as a ground filter was compared to grids of manually interpreted points in section 2.5.2.

The errors found in the manual inspection were mostly caused by industrial roofs and dense forests. Larger errors (caused by omission of entire hills) were only noticed twice. Outside these errors the interpreters felt that the GEM was largely accurate, containing negligible errors.

The surface did not appear affected by outliers below or above the true DSM. The only cases of DSM errors affecting the GEM occurred where entire frames of data were corrupted. This error tolerance in the GEM meant that the difference between the DSM and the GEM was actually very useful for detecting errors in the DSM.

The GEM was accurate for most roof types, including complicated roofs,

however it typically was incorrect over large industrial roofs such as saw-toothed roofs. This was because inflows was confused by their unusual shapes, and their large areas protected them from the surface-based filter.

Objects attached to the ground were often labelled as bare earth, however this proved to be a rare issue generated mostly by rooftop car parks on buildings in commercial regions. It is uncertain whether bridges should also be classed as attached objects because in some real-life cases they are regarded as ground. In any case, our GEM partially included most bridges as ground; inflows labelled them as ground and the smoothing filter eroded the edges.

Vegetation caused big issues only in dense forests where the amount of erroneously commissioned pixels from inflows was usually very small, but the small amount of clear, open ground meant that the surface fitting could not remove them.

As expected, discontinuities in the true ground were smoothed over, along with gullies and the peaks of sharp dunes.

2.5.1 Time Required

The final, high quality surface fit required a total of 49.9823 work units. For a grid of 182.96×10^6 pixels it took 1059 seconds[6] to complete on a 3.10 GHz Intel i5 CPU running 32-bit Windows 7 with 4GB of RAM. A work unit was thus equivalent to 21.19 seconds of computation, suggesting a time of 0.1164 seconds per megapixel per work unit.

The total work units required for creating a GEM (outlier removal and final surface fit) in the Darling Ranges and coastal plain was 75.8001 and 73.3587 respectively. Thus on this particular computer it would respectively cost 8.823 and 8.539 seconds per megapixel to generate a GEM from the candidate ground mask.

If the GEM for the entire Urban Monitor region was calculated using this desktop computer it would take 24 days (ignoring the need to perform the fitting piecewise), in practise we used a larger and faster supercomputer (see chapter 4).

[6]this is from just one execution of the algorithm and the time taken could have been affected by other processes running on the computer - for a better idea of the speed many completetions of the algorithm could be assembled into a distribution.

Note that on this machine only input data with 1.87×10^8 pixels (748MB) or less could be used.

2.5.2 Accuracy

In clear, flat, open ground, the GEM was nearly identical to the DSM and consequently inaccuracies of the GEM were attributed to two sources, the input DSM and any commission/omission errors of the ground filter (a third source, occlusions, was also possible in very dense forests where there was very little visible ground). For this investigation we have considered ground commission/omission errors and gross DSM errors only.

The manually interpreted grids from section 3.1.2 were compared to a ground/non-ground classification generated by thresholding the 2007 nDSM at 300mm. These manually interpreted points were spread across three small regions on the coastal plain. See table 2.3 for the confusion matrix. An overall accuracy of 89.6% was achieved with a 2.2% commission error and a 8.1% omission error. This commission error rate was only partially due to the GEM algorithm, many errors were due to DSM errors such as matching issues in or near trees (e.g. where the DSM completely missed a tree). The high omission error rate (a quarter of predicted non-ground pixels were labelled ground in the manual interpretation) was also due to more than just the GEM algorithm. Some of these points were erroneously predicted as *non-ground* due to smoothed object edges in the DSM, stereo-matching errors in trees or errors caused by moving objects.

Due to time constraints no manually interpreted points existed to quantitatively test the accuracy in the Darling Ranges. The steeper terrains and different parameters may have resulted in poorer accuracy, although the manual inspection showed that the GEM algorithm still functioned well.

It is difficult to compare these accuracies to other GEM creation methods because the complexity of the landscape, the resolution and the sensor type all play a significant role. The very limited access to other algorithms meant that it was only possible to apply one other algorithm (from the Inpho program) to the same input data as our GEM algorithm. Furthermore due to time and data transfer limits it was only possible to compare algorithms on a small region (see the following section for the comparison between Inpho and our algorithm).

predicted	manual interpretation			
	ground	non-ground	unknown	sum
ground	922	32	56	954
non-ground	116	354	91	470
sum	1038	386		1424

Table 2.3: Confusion matrix for the 2007 GEM over the three sites in section 3.1.2. The proportion of predictions that matched the manual interpretations (the overall accuracy) was 89.6%, the percentage of manually interpreted pixels that were *incorrectly* predicted as *ground* (the commission error) was 2.2% and the fraction of manually interpreted pixels that were *incorrectly* predicted as *non-ground* (the omission error) was 8.1% (these accuracies and the sums of each row above are ignoring the unknown pixels).

2.5.3 Comparison to an Inpho match-T GEM

Inpho's match-T algorithm [62] was used to create both a DSM and a GEM for a $4.5km \times 5.7km$ region representing a fairly simple, but common landscape in the Urban Monitor region. The region was on the flat coastal plain and contained clear ground, sparse forest and large, highly vegetated residential blocks.

Images of the results are shown if figure 2.10. Exactly the same raw photographs from the 2009 data were used for both algorithms. The default parameters were used for the Inpho algorithm.[7]

In clear open ground both GEMs closely matched the DSMs so numerical height accuracies were not compared. However there were clear differences in the abilities of each method to filter out non-ground objects.

Each ground elevation was evaluated by manually scanning through the region at high resolution, noting any commission or omission errors. For an error to be noticeable it usually needed to be an area of approximately $300m^2$ or larger where the GEM was either rougher than the user expected (e.g. the surface contained objects, protrusions or small hills where nothing in the DSM or orthophotos indicated a change in height), or was smoother than it should be (e.g. loss of gullies and sharp landscape features).

The height of the two GEMs were then compared to highlight further differences, this was especially useful for discerning omission errors in our GEM.

[7]Our thanks to Aaron Thorn for generating the Inpho GEM.

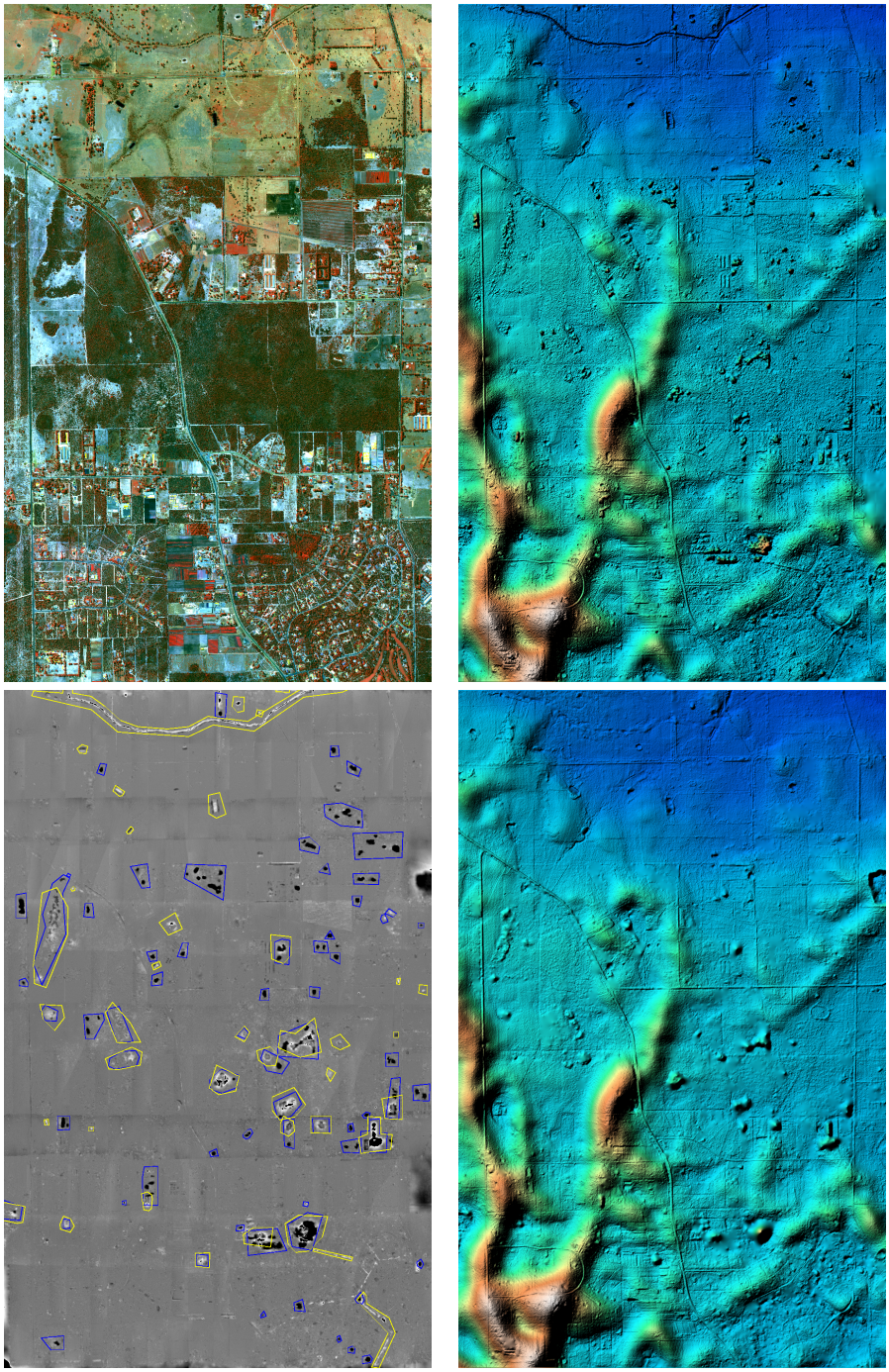


Figure 2.10: Comparison with the match-T generated ground elevation. Shown is a false colour multispectral display *(top left)*, with the GEM generated by Inpho *(top right)*, our GEM *(bottom right)* and a grey scale difference of the two GEMs with the errors noted in polygons *(bottom left)*. Many more errors were found in the Inpho GEM (blue polygons) than our GEM (yellow polygons). Note that the Inpho GEM contained some matching issues at the edge of the region which were ignored.

Differences that the user could not easily determine as an error were ignored.

A total of 40 errors were found in our GEM. There were 20 commission errors of trees, 15 omissions of dams and gullies, 3 roofs commissioned as ground and 2 incidences where the GEM smoothed over true bulges in the ground.

A total of 67 errors were found in Inpho's GEM. There were 59 errors caused by trees, 6 roof errors, and 2 erroneous protrusions in the DSM that the GEM failed to remove. No omission errors were found; it did not smooth over gullies or dams.

Whilst scanning the Inpho GEM it was also noticed that the surface contained some tiling issues in areas where the ground was occluded (see figure 2.11). This lack of smoothness will be troublesome for use with flow modelling or watershed type algorithms.

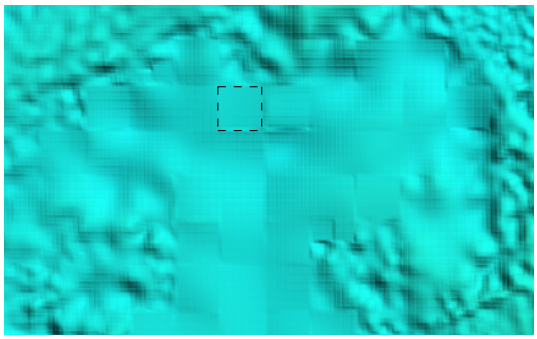


Figure 2.11: Example of the Inpho GEM tiling issue. Shown is the sun-shaded Inpho GEM surface. Tiles 50 pixels wide are apparent (dashed square), revealing the piecewise nature of the surface fitting.

Overall Inpho's GEM performed relatively poorly in vegetation and roofs, with almost three times as many vegetation commission errors as our GEM, and twice as many roof errors. However it performed much better around sharp changes in the ground which were mostly caused by gullies and dam walls. Our GEM consistently smoothed over these sharp changes.

In dense urban areas and forests match-T is likely to create many more roof commission errors, and in some locations it will probably generate surfaces at roof or canopy height because it only filters points based upon whether they are outliers from a fitted surface [62]. Whilst improvements of Inpho in these areas is

almost certainly possible by using different parameters, it is difficult to see how it could better our GEM without many stratifications of parameters.

2.6 Conclusion and Discussion

In this chapter we developed a new hybrid ground filter that combined a segmentation/morphological based filter and a surface-based filter. The combination arose naturally from the small yet significant amount of commission errors made by the segmentation filter.

The manual investigation of the GEM showed the algorithm functioned extremely well in most terrains in the Perth area: well-vegetated dense suburbs, hilly suburbs, new suburbs, peri-urban areas and farmland. That only two different parameter sets were required to achieve these results attests to the quality of the algorithm. However it failed on a few rarer terrains: industrial buildings, dense forests, extremely steep hills, sharp depressions and discontinuities. Affects of these errors on applications will depend on what is being modelled, the resolution of the models and the error locations.

In a simple peri-urban region it easily out-performed the commercial match-T algorithm and it is expected that the improvement over match-T will be more significant in regions with more aboveground objects such as in dense urban areas. The accuracy when considering the nDSM as a ground filter was 90%, although accuracy could be poorer in the Darling Ranges.

Comparing the accuracies and quality of GEM generation algorithms is currently quite difficult. Presently comparisons to widely distributed commercial programs are the only option (without significant collaboration), however this does not allow for quantitative comparisons between newer (or non-commercialised) algorithms. In the future this could be solved by a collection of publicly available datasets containing a representative sample of landscapes, resolutions and sensor artefacts (similar to the temporarily available datasets used in [85]) that each researcher could apply their own algorithm to.

Development of the surface fitting algorithm suggested that adaptive multigrid cycles were not ideal, and the much simpler cascadic design proved sufficient for the GEM generation.

The generation of a GEM is essential to many image recognition algorithms.

It allows for detection of the height of non-ground objects, such as the heights of trees and buildings through the nDSM. It can also be used in a wide range of other fields, such as flood modelling. Combined with a classification scheme (such as in chapter 3) it leads immediately to simple environmental indicators such as the proportion of vegetation above five metres.

For convenience in this study the candidate ground points were cleaned according to roughness. However many different filtering methods are possible after the surface fitting. Other cleaning methods that have been used in a similar context are a threshold on the residual of candidate ground points [21] or the morphological top-hat transform [67]. A novel suggestion would be to reuse inflows by applying it to the DSM normalised by an extremely smooth surface fit.

Manually interpreted points in the Darling Ranges are needed to determine the accuracy for that region and if there were manually interpreted points spread evenly across the entire Urban Monitor region then more representative accuracies overall could be generated.

Whilst the surface fitting could cope with discontinuities they required a reliable discontinuity detection system. The variety of landscapes and the presence of occluded discontinuities (such as tree crowns above cliffs) made any such system too time consuming to implement. In the future with discontinuities incorporated into the GEM it will be possible to correctly fit gullies, terraces and other rough features, and thus avoid many omission errors.

Chapter 3

Classification

Classification of land cover from aerial and satellite images is vital to many automatic environmental remote sensing objectives. It remains an ongoing field of research, in part due to the complexity of landscapes and increasing capabilities of sensors.

An algorithm that assigns pixels to categories (or classes) is known as a *classifier* and the output is known as a *classification* or a *thematic map* (because the classes are often relevant to specific themes). The indices used to discriminate pixels into different categories are often called *features*.

Land cover classifications can be used to indicate the health of ecosystems through structural vegetation characteristics and the analysis of dynamic environmental systems such as energy balance, biogeochemical cycles and hydrological cycles [27]. Automated classifiers allow this sort of analysis to occur at high resolutions, over entire cities and for multiple dates.

Some thematic maps that have received attention in remote sensing are impervious surfaces [86], land cover changes [27, 77, 41], building footprints [23, 50], avian habitats [70, 43], vegetation maps [7, 9, 104], individual tree maps [26, 69, 60] and individual species maps [35].

Classifications are also required for more advanced comprehension of images [94]. Towards this end much research has been conducted into classifying urban areas into the common cover types such as shadow, grass, tree, roof, road, pavement, bare soil and water [81, 48, 8, 52, 51, 102, 34].

Lu and Weng [64] provided a synopsis of the types of classifiers used for remote sensing image analysis, listing a number of different descriptors that can

be attached to classifiers:

- Supervised. Classifiers that are trained on a set of samples are known as *supervised*. If no training samples are used (e.g. in some clustering algorithms) then it is an *unsupervised* classifier.

- Parametric. A parametric classifier models the data as a particular distribution (e.g. a normal distribution).

- Per-pixel, object-oriented or per-field classifiers. A classifier that labels a single pixel at a time is *per-pixel*, otherwise it labels groups of pixels at a time. If the groups are created using some object recognition or image segmentation approach then it is as an *object-oriented* classifier. If the groups are generated from a GIS database (e.g. a map of land ownership) then it is known as a *per-field* classifier.

- Hard or soft (fuzzy). A *hard* classifier assigns each pixel to precisely one class. Conversely if pixels are assigned probabilities or a membership value of classes then it is a *soft* or *fuzzy* classifier.

- Point-based or contextual features: The features used by the classifiers can be thought of as either *point based* (use information from a single location) or *contextual* (use spatially neighbouring information). Classifications may use only one feature, or can combine many features (e.g. [52] used seventy-eight features for one classifier).

Apart from design, other factors that influence the quality of a classification are the resolution and quality of the data, and the complexity of the region.

The remote sensing image analysis field owes many techniques to the more general field of computer vision and pattern recognition. However the greater a priori knowledge of sensors and objects within the (Earth) remote sensing images allows for many unique methods. One of the earliest and most popular examples of this is the *normalised difference vegetation index* (NDVI) [78] which is based on the reflectance of chlorophyll [7] and a need to normalise illumination differences.

In this chapter a method for deriving point-based features and indices, *Canonical Variate Analysis with Rational Polynomials* (CVAR) [28], is investigated

along with the use of NDVI for generating vegetation maps from the Urban Monitor data.

CVAR was used to generate multiclass (four to eight classes) thematic maps with estimated accuracies greater than 77% in three regions between $5 - 16km^2$ in area. CVAR was not used to create multiclass classifiers for the full Urban Monitor region due to time restrictions and the need to stratify the region. However it was shown that a two-class vegetation/non-vegetation classifier using the NDVI functioned well over the entire area, requiring only a few strata.

CVAR combines standard canonical variate analysis (CVA) [19, 18] with a normalising denominator. The denominator is intended to reduce the *topographic illumination effect*, which is the variability in reflected light caused by surfaces at different angles. The use of ratios to reduce the topographic effect in indices is well established [47, 7] but the optimisation of the denominator is rare.

Other strategies for removing the topographic effect require a DSM and simulate the illumination conditions, but do not always perform well [76]. A benefit of CVAR over these methods is that it searches for the best topographic effect correction for the separation problem, rather than trying to perfectly remove the topographic effect in every pixel. It also does not depend on the accuracy of the DSM.

In section 3.1 we examine CVAR with various experiments, including the generation of some hierarchical classifiers, finishing with some guidelines for further use (subsection 3.1.6). The development of the vegetation classifier that was based mostly upon NDVI and assisted by a morphological object-oriented feature is described in section 3.2.

3.1 Classification by Canonical Variate Analysis with Rational Polynomials

Canonical Variate Analysis with Rational polynomials (CVAR) is a method for discriminating classes in areas effected by topographic illumination differences without the need for an elevation model. It was first introduced by Chia et. al. [28] where it was tested on Landsat TM data. Terrain illumination correction is especially important for high resolution aerial photography because

- there are many non-ground objects with many different surface angles,
- the illumination direction and viewing angle changes are more frequent due to the larger number of frames captured over many hours and days[1], and
- the field of view (or viewing angle) of airborne sensors is much larger.

Standard CVA [19] finds the linear transform $c \cdot x$ of input variables x that maximises the variation between groups whilst minimising the variation within groups. CVAR generalises this to a transform $\frac{c_1 \cdot x}{a + c_2 \cdot x}$ (a is a fixed constant). This ratio transform is based upon the well known concept that a ratio of bands reduces the effects of terrain illumination (e.g. [78, 66, 103, 47]).

The results of CVAR may be applied in exactly the same fashion as CVA. They can be used to explore the data, derive indices for immediate use, or create features for use in classification. The benefit of CVAR is that if the training data is chosen well then the transformed values are less affected by surface angle differences, generating much cleaner separations between classes.

The theory of ratioing to reduce the topographic illumination effect is described well in [47]. The light received by a sensor can be modelled using a multiplicative term and an additive term. The additive term is due to ambient light hitting the sensor whilst the multiplicative term is a combination of solar angle and intensity, reflectance of the target (angle dependent) and atmospheric transmittance (also angle dependent). Elimination of the topographic effect via ratios relies on two main assumptions (1) that the intensity of ambient light is insignificant (or already removed) and (2) the angle dependence is the same for every wavelength. When these assumptions are true then a ratio of two different bands (e.g. nir/r or ratios of any linear combination of bands) will have the same value for a material irrespective of angle. In practice these assumptions do not always hold and results can be further degraded by preprocessing errors (for example the Landsat calibration and integer quantisation [47]). The optimisation of the ratio in CVAR is an acknowledgement of these broken assumptions and imperfections.

[1] although greater control over aircraft compared to satellites means the illumination angle can be controlled to some extent

Apart from CVA other examples of supervised feature extraction methods are principle component analysis (PCA) and maximisation of mutual information [58]. CVA is a general tool for separating groups of data points and is one of the most widely used multivariate statistical techniques in biology [19]. It has been used in remote sensing for image fusion [72], vegetation classification [45] and dimensionality reduction [95].

We describe CVAR in more detail in section 3.1.1. Tests into the ability of CVAR for designing multiclass classifiers for large areas ($5-16km^2$) are presented in section 3.1.2. The separations from CVAR were shown to outperform CVA and a more general version of CVAR in most cases (section 3.1.3). A demonstration of the reduction in topographic effect from CVAR is included in section 3.1.4. Experiments in the numerical search of c_2 (section 3.1.5) showed that up to a size limit the results were usually independent of the initial guess for the numerical search. A collection of guidelines for the use of CVAR is included in section 3.1.6.

3.1.1 CVAR Details

Suppose we have a set of training data $\mathcal{D}$, sorted into g groups $\eta_1, ..., \eta_g$ and have chosen a number a. The aim of CVAR is to find two vectors c_1 and c_2 such that

$$\mathcal{D}_{transformed} := \left\{ \frac{c_1 \cdot x}{a + c_2 \cdot x} : x \in \mathcal{D} \right\}$$

obtains the maximum possible ratio of between-group variation to within-group variation. Where the variations are calculated using the univariate versions of the between-group sum-of-squares-and-products matrix

$$\mathbf{B} = \sum_{k=1}^{g} n_k \left(\bar{x}_k - \bar{x}_T\right) \otimes \left(\bar{x}_k - \bar{x}_T\right) \tag{3.1}$$

and within-group sum-of-squares-and-products matrix

$$W = \sum_{k=1}^{g} \sum_{m=1}^{n_k} \left(x_{km} - \bar{x}_k\right) \otimes \left(x_{km} - \bar{x}_k\right) \equiv \frac{1}{n-g} \sum_{\eta=1}^{g} (n_\eta - 1)\text{covariance}(X_\eta) \tag{3.2}$$

where n_k is the number of observations in the k^{th} group, x_{km} is a vector of the values of the m^{th} observation in the k^{th} group, $\bar{x}_T$ is a vector of the means for

the entire data set, $\bar{x}_k$ is a vector of the means for the k^{th} group, and $\otimes$ denotes the outer product between two vectors.

We refer to this maximum ratio of between-group sum-of-squares to within-group sum-of-squares as the CVAR *canonical root* of $\mathcal{D}$.

For a given c_2 the optimal c_1 is simply the solution to the standard CVA problem but with the data first transformed to $\frac{x}{a+c_2 \cdot x}$. Much of the theory for CVA is described by Campbell and Atchley in [19] where an equation is given for calculating the optimal transform vector c_1 and a measure of the separation quality (the *canonical root*) directly from an eigenvector analysis. This calculation uses the within-group $\mathbf{W}$ and between-group $\mathbf{B}$ matrices of the initial data $\mathcal{D}$ (see the section on CVA functions in appendix D.1.1 for full details). Following the methods in [20, 17] improvements to stability were gained through standardising $\mathbf{W}$ and $\mathbf{B}$ such that $\mathbf{W}$ had unity on the diagonals and removing directions with extremely low within-group variance.

Solving this eigen-analysis problem returns the best c_1, other orthogonal vectors (orthogonal with respect to the within-group variance), and the canonical roots for all of these vectors. These vectors are known as the first, second, third... *canonical vectors*, (denoted as $c_{1a}, c_{1b}, c_{1c}, ...$ in this thesis) in descending order of canonical root. Consequently the transforms $\frac{c_{1a} \cdot x}{a+c_2 \cdot x}, \frac{c_{1b} \cdot x}{a+c_2 \cdot x}, \frac{c_{1c} \cdot x}{a+c_2 \cdot x}, ...$ are referred to as the *CVAR transforms*.

The optimisation of c_2 was performed by the Nelder-Mead Simplex Search Algorithm [68], the objective function being the CVA canonical root of the data transformed by $\frac{x}{a+c_2 \cdot x}$. The Nelder-Mead algorithm is an extremely popular non-linear unconstrained optimisation algorithm [56]. See appendix D.1.1 for the details and parameters used.

The constant a was chosen to be non-zero for stability reasons (confirmed in section 3.1.5), $a = 0$ frequently resulted in division-by-zero errors. The choice of a is otherwise irrelevant because a factor of $1/a$ can be factored into c_1 and c_2. For simplicity we always set $a = 1$.

In our implementation the initial simplex guess was chosen randomly by selecting each ordinance from a uniform distribution between $[-1, 1]$. The entire simplex was then scaled by a factor determined by the user. Experiments (section 3.1.5) showed that the size could be chosen such that the shape of the simplex had little impact on the final result.

The functions for importing training data, performing CVA and the simplex search, and plotting the results were mostly written in the statistical computing environment/language R [73]. Details of these functions can be found in appendix D.

3.1.2 Experiments in Multiclass Classifications

CVAR was used to generate multiclass classifications in a hierarchical fashion for three large pilot regions in the 2007 Urban Monitor data. These studies confirmed the usefulness of CVAR canonical vectors for classification, investigated the ability of the classification rules to generalise to the entire Urban Monitor region and revealed many issues and best practises for applying CVAR.

The classification in one region was able to extract nine spectral classes such as concrete, water and shadowed vegetation with an estimated accuracy of 77.5%. However the spectral mosaic just for this study was stitched together from a small number of frames which gave a much more consistent and uniform calibration than the full 2007 Urban Monitor mosaic.

The other two studies used data extracted from the full 2007 Urban Monitor mosaic and their classification rules could be tested over the entire region. Investigations showed that generally the accuracy spatially varied significantly.

Overall CVAR performed poorest when distinguishing shadow and water from other cover types.

Description of Regions

The sites were chosen according to the complexity of the region, environmental importance and differing soil types. They were all on the coastal plain of Perth (see figure 1.1), and contained a range of suburbs, landscaped parks, native bushland and farmland. False colour displays of all the sites are shown in figure 3.1. Subiaco-Kings Park was the smallest region at only $1.4km \times 3.5km$ or 125×10^6 pixels, all the other regions were greater than $3.2km \times 3.2km$.

Region size is only a surrogate for complication and variation in the data; as the area increases typically more cover types, spectral variation, unusually shaped objects and errors occur. One site, Forrestdale, was particularly complicated containing large shadows from clouds, burnt forests, burnt grass and a large dry

lake.

The first pilot study was conducted on the Subiaco-Kings Park area (shown in figure 3.1a). This area ($1.4km \times 3.5km$) contained many commercial buildings, old, well-vegetated residential suburbs, landscaped gardens, a railway reserve, and some native bushland. The frames used for this study (fewer than one thousand) were calibrated and mosaicked separately to the rest of the Urban Monitor data. As a result the training pixel values in the study were more consistent than the values in the other pilot studies. CVAR was able to separate many more classes in this region than in either of the other regions.

The Forrestdale and Jandakot Airport sites were combined to create the second pilot study (shown in figures 3.1b, 3.1c respectively). The two regions were chosen for their importance to local environmental agencies. The Forrestdale area ($4km \times 4km$) was almost a quarter farmland and contained significant amounts of bushland, a small residential estate, an environmentally fragile dry lake, recently burnt vegetation around the edge of the lake, and shadows from clouds. These complications made it extremely difficult to classify by automatic methods. Jandakot airport ($3.5km \times 3.5km$) contained sparse native forests, sparse residential suburbs, and an airport along with commercial buildings. The two areas were combined for training purposes because they were only a few kilometres apart, and it was difficult to find shadow training sites within Forrestdale alone.

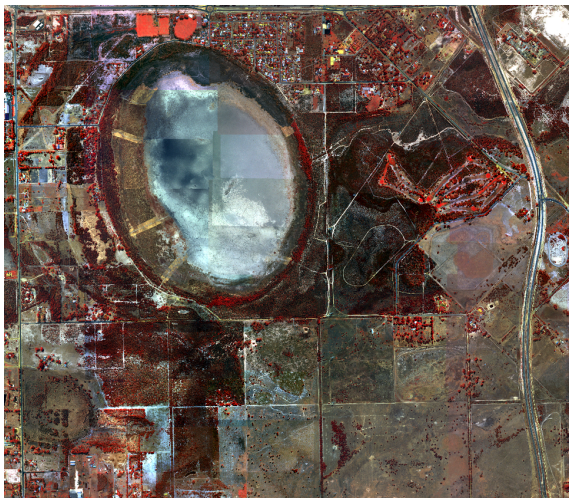
The Rockingham pilot study ($3.2km \times 3.2km$), shown in figure 3.1d, was chosen as a similar landscape to Subiaco-Kings Park, but with a different underlying soil type [33], less vegetation and less crowded suburbs. It contained residential buildings, commercial buildings, low lying native scrub and some ocean.

Methods

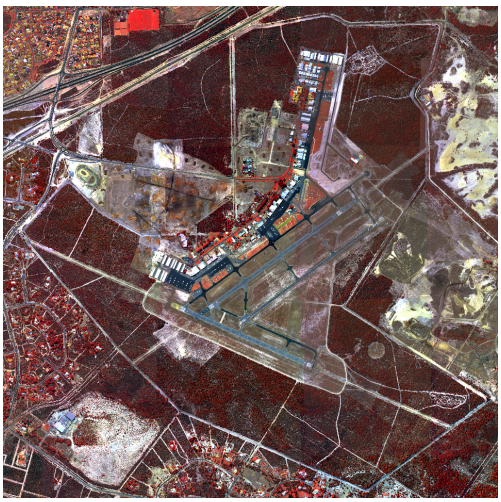
For each pilot study the training data was a collection of small rectangular regions (sites) of homogeneous cover, each labelled according to one of twenty or more descriptive types such as concrete, irrigated grass, non-irrigated grass, bright swimming pools, synthetic green grass, shaded irrigated grass, shaded concrete etc. The sites were found by methodically scanning the region by eye and care was taken to keep their size approximately constant.

Multiclass classifiers were created by generating discrimination functions in a hierarchical manner. At each stage the following procedure occurred:

(a) Subiaco-Kings Park



(b) Forrestdale



(c) Jandakot Airport

(d) Rockingham

Figure 3.1: The CVAR training regions. For the locations within the Urban Monitor region see figure 1.1.

1. CVAR was applied to the training data. Usually a number of different CVAR variations were attempted before good separations were discovered. The size of the initial simplex in the simplex search was usually 10^{-4} however when this size failed other sizes were tested. The results of CVAR were investigated through 2D plots of the training pixels transformed by the first two CVAR canonical transforms. The order of attempts to find separations was:

   (a) A site-wise CVAR on the full training data: each site was considered a unique group (this option was not used in Subiaco-Kings Park). In other words CVAR searched for the maximum distance between sites, whilst keeping each site as compact as possible. This allowed for easy identification of issues in the training data such as extreme sites and pixels, sites with large variations, and mislabelled sites. Usually CVAR was reapplied ignoring these problematic sites which led to better separations with more discriminatory CVAR transforms. The ignored sites were often included in the 2D plot to ensure that any discrimination boundaries were drawn to correctly label all the sites.

   (b) CVAR was performed with groups given by the fine-grained classes used to label the sites. Again extreme sites/groups were ignored (including any found in the previous site-wise attempts).

   (c) Directed contrast CVAR: Guided by the previous attempts broader groups were chosen (usually the classes that were desired for the final classification, such as 'brown cover' or 'green vegetation'). By applying CVAR in this manner the greater emphasis on separation between the broader types would sometimes yield enough difference to draw a reasonable boundary. Again extreme sites/groups were ignored.

   (d) Finally an even more directed contrast was attempted: Here multiple broad cover types were merged. For example, an attempt to separate green vegetation could use a two-group CVAR where one group contained the green vegetation pixels and the other group contained every other training pixel. Again extreme sites/groups were ignored.

2. Once a satisfactory separation was found, boundaries (thresholds) were

drawn around the distinct cover types to form discrimination functions from the canonical transforms.

3. The pixels corresponding to the discriminated class(es) were removed from the training data. The remaining training pixels were used to generate separations in the next stage.

The final classes from the hierarchical CVAR classification were then further separated into ground and aboveground types by thresholding the nDSM (generated using methods in chapter 2) at 300mm.

The classifications were compared to regular grids of manually interpreted pixels in Rockingham, Forrestdale and Subiaco-Kings Park. The manual interpretations were performed using the calibrated orthophotos. If it was difficult to determine the membership of the point then it was put into the Unknown class. If an obvious class did not exist because it was an unusual or rare cover type (e.g. car windscreens) then the point was put into the class Other.

Results

In general the results of the accuracy analysis are presented in raw confusion matrices (see table 3.1 for an example) which is a common recommendation for analysing the accuracy of thematic maps [30, 39]. The sums of each column and row, and the producer's and user's accuracies are also given. The producer's accuracy [30] of a class, $\mathcal{A}$, is defined as the proportion of the points manually labelled $\mathcal{A}$ that were interpreted as $\mathcal{A}$ by the automated process:

$$\text{producer's} = \frac{\left|\mathcal{A}^m \cap \mathcal{A}^a\right|}{\left|\mathcal{A}^m\right|}.$$

Where $\mathcal{A}^m$ is the set of points manually labelled $\mathcal{A}$ and $\mathcal{A}^a$ is the set of points labelled $\mathcal{A}$ by the automated process ($\cap$ is the intersection operation and $|\mathcal{X}|$ the number of elements in the set $\mathcal{X}$). Thus the producer's accuracy is an indication of the completeness of the automated class label.

The user's accuracy [30] of a class $\mathcal{A}$ is the fraction of the points automatically labelled as $\mathcal{A}$ that were manually interpreted as $\mathcal{A}$:

$$\text{user's} = \frac{\left|\mathcal{A}^m \cap \mathcal{A}^a\right|}{\left|\mathcal{A}^a\right|}.$$

Where $\mathcal{A}^m$ is the set of points manually labelled $\mathcal{A}$ and $\mathcal{A}^a$ is the set of points

labelled $\mathcal{A}$ by the automated process. The user's accuracy is thus an indication of the probability that an automatic label of $\mathcal{A}$ is correct.

The overall accuracies were the sum of the correctly labelled points over the total number of points.

The points in the Unknown class (coloured grey in the confusion tables) were not used in the accuracy statements.

Subiaco-Kings Park. In Subiaco-Kings Park a total of nine different classes were separated over six stages. Site-wise analysis was not used in this study.

1. The classes pools, synthetic green cover and green vegetation cleanly separated using a directed CVAR comparison between the groups pools, synthetic green cover, green vegetation, and an everything else group (this last group contained all the training pixels that did not match the former three groups). See figure 3.6 (centre) for the plot of the training pixels transformed to the first two CVAR transforms.

2. The orange-brown class separated next, with a comparison between the groups shadowed vegetation, orange-brown and {grey-white}∪{black tar}∪{concrete}∪{shadowed non-vegetation}.

3. Grey-white cover was then separated from the remaining training pixels using a comparison between the groups: grey-white, concrete, shadowed vegetation and {black tar}∪{shadowed non-vegetation}.

4. Concrete marginally separated next using a two group analysis between concrete and {black tar}∪{shadowed non-vegetation}∪{shadowed vegetation}, leaving black tar, shadowed vegetation, and shadowed non-vegetation to be discriminated.

5. Black tar was separated using another two-group contrast, and finally

6. Shadowed vegetation was separated from shadowed non-vegetation with yet another two-group contrast.

The manual interpretation on the grid revealed a significant new cover type that was not in the training data: dry grass. For the accuracy analysis (see table 3.1) dry grass was considered part of the brown class. In practise dry

grass and dirt are very spectrally similar (no classifier investigated in this thesis distinguished them) however their differentiation in the future would be quite useful for fire modelling, habitat analysis/monitoring, and many other topics.

The spectral-only classification achieved an overall accuracy of 77.5% (see error table 3.1). However concrete, pools, synthetic green cover and the shadow classes were not sampled enough to provide meaningful accuracy statements. Manual inspection of the thematic map showed that these classes were extracted reasonably well except for the discrimination between the two shadow classes (for an example see figure 3.2). Some representative examples of the full classification are shown in figure 3.3.

The largest confusion in the common cover types was between black tar and grey cover. For an example see figure 3.3d, where most of the black tar roads were labelled grey ground. All the common cover types achieved user's and producer's accuracies of at least 70%. Green (photosynthetically-active) vegetation was classified the most accurately with user's and producer's accuracies of 96.5% and 88.2% respectively.

For completeness an accuracy summary of the thematic map separated by a $300mm$ nDSM threshold is presented in table 3.2. Due to low sample rates, some of the classes have been merged together for this confusion table. Because of this the overall accuracy, 77.4%, does not reflect the errors introduced by the nDSM. From the table it can be seen that there was large confusion between green grass and green trees, and similarly between grey ground and grey roofs.

The quality of this classification across such a large area, even within shadows is a testament to both the illumination normalisation of CVAR and the quality of the calibration technique.

Jandakot and Forrestdale. Two different classifiers were trained on this region, one that only used site-wise separations and another that only used directed contrasts. In both versions four classes, brown, grey, green (photosynthetically-active) vegetation and shadow were discriminated.

Site-wise separations: In the site-wise separations many extremely bright brown or white sites were removed first and then extremely bright green vegetation pixels were separated. Once all of these extreme bright sites were removed, the

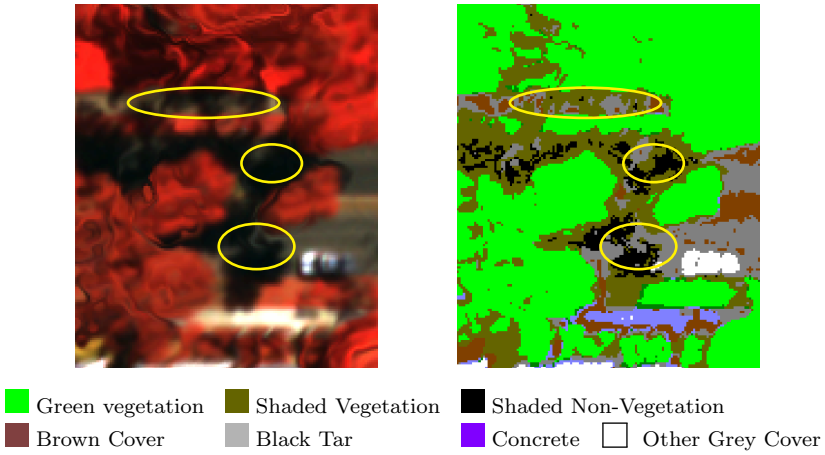
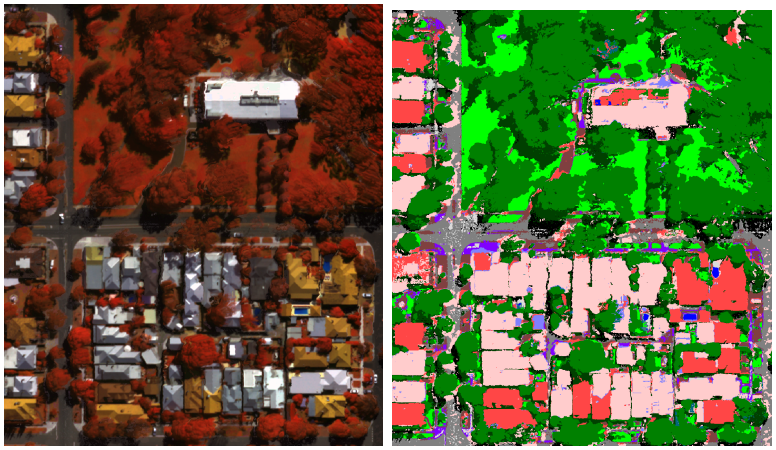


Figure 3.2: The discrimination of shadowed vegetation and shadowed non-vegetation. A false colour image (left) with the spectral-only classification (right) is shown. The separation was much poorer than the other classes; in this example many of the shadowed road and pavement pixels have been labelled shadowed vegetation. Some regions of particularly certain errors (often it is difficult to manually discern whether shadowed pixels are vegetated) have been circled in yellow.

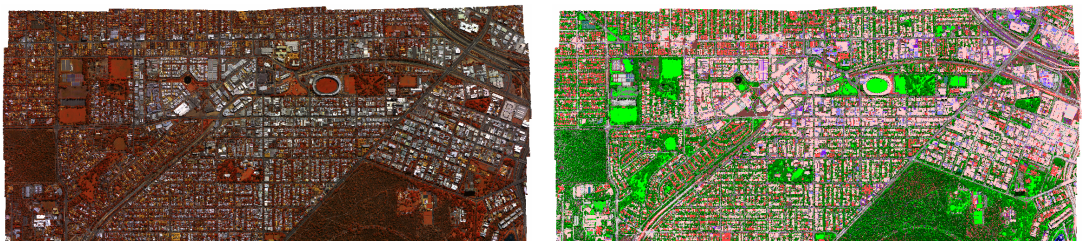
normal grey sites were separated. In the next stage shadows separated from the rest of the sites. Finally the normal vegetation sites and normal brown sites were separated from each other.

Many of these separations were dominated by illumination differences (see figure 3.4) and often it was impossible to find boundaries that correctly assigned all of the training pixels. Despite this the accuracies of many of the classes were higher than 70% (table 3.3). The producer accuracy for shadows and the user accuracy for grey cover were the exceptions. More than half of the manually designated shadow pixels were not shadow according to the classifier, whilst many brown pixels were labelled grey by the classifier.

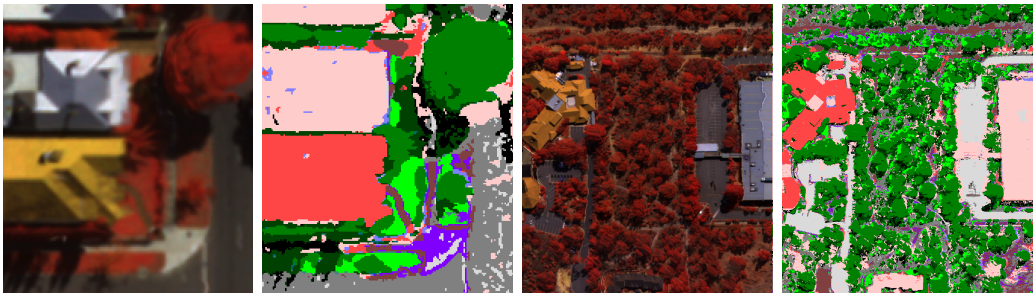
Directed contrasts: The classifier generated using directed contrasts produced the same classes and the same overall accuracy (table 3.4) as the site-wise classifier. It managed to correctly label 71.4% of the shadow pixels, but committed almost as much brown and green vegetation as shadow as there were true shadow pixels.



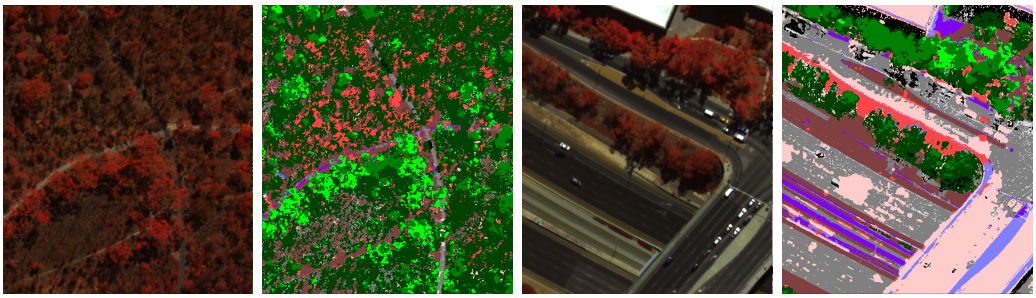
(a)



(b)



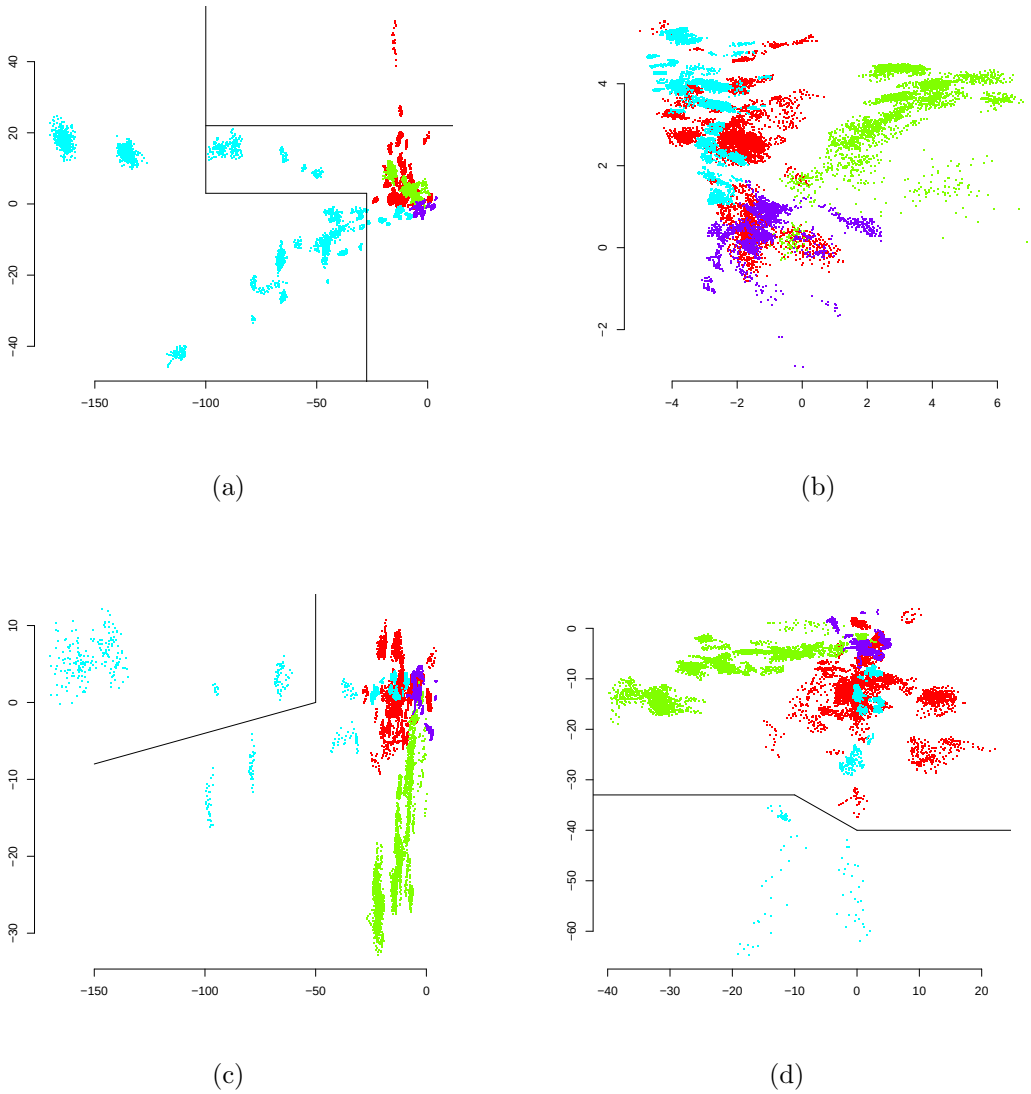
(c) (d)



(e) (f)

Green Grass, Green Tree, Synthetic Green, Shaded Vegetation, Other Shadows
Brown Ground, Black Tar (road), Concrete Pavement, Other Grey Ground
Brown Roof, Grey Roof, Concrete Roof, Pools

Figure 3.3: Examples of the classification of Subiaco-Kings Park.



(a) (b)

(c) (d)

Figure 3.4: Example CVAR separations. Shown is the first site-wise separation in Forrestdale-Jandakot *(a)*, a directed contrast of the same training data *(b)* (the groups in this directed comparison were other, green vegetation, brown vegetation and shadow), and the second *(c)* and third *(d)* site-wise separations used in Forrestdale-Jandakot. The plots show training data transformed by the first two CVAR canonical transforms. Each training pixel is coloured according to a type: white-grey, orange-brown, green vegetation, or shadow. The discrimination boundaries used for the site-wise Forrestdale-Jandakot trained classifier are also shown (black lines). The dominance of brightness in the site-wise separations is evident from the clustering of most sites (including all the shadows sites) and the rays of brighter sites emanating from this cluster.

It was hoped that a greater use of site-wise comparisons would lead to classifiers that produced good results over larger areas. To this end the classifiers were applied to the Rockingham and Subiaco-Kings Park regions. Both classifiers performed similarly in Rockingham (table 3.5) with both achieving very poor brown user's accuracy and very poor shadow accuracies (although not enough shadow pixels were available to give a definitive result for the latter). In Subiaco-Kings Park the directed classifier was slightly superior with better user and producer accuracies for every class except for brown which had a poorer user's accuracy. The site-wise classifier had a particularly poor producer's accuracy of 30.8% for shadow.

Despite the high overall accuracies in Subiaco-Kings Park (more than 80%) visual inspection revealed that neither classifier worked as reliably in areas closer to Forrestdale and Jandakot, in fact their accuracies appeared to vary every few image frames.

Rockingham. The Rockingham separation proceeded similarly to Forrestdale by using site-wise comparisons and removing any extreme training data. Six classes separated spectrally: shadow, brown, grey, green (photosynthetically-active) vegetation, bright water and blue roofs. Initially there was a problem with a massive ocean training site skewing the CVAR results, however once the size was reduced different cover types started separating.

As with the site-wise Forrestdale separations the classes were extracted using a number of stages, with most classes requiring two or more stages. For example the brown cover was separated over three different stages, first the bright brown pixels, then the more normal brown pixels and finally some grey-brown pixels. This staged removal may be thought of as extracting subtypes of brown pixels although it could also be an artefact of the training data.

Shadows, blue cover and bright water were not sampled sufficiently to give reliable accuracies. The other three classes brown, grey and green vegetation all achieved greater than 69% user and producer accuracies.

The classifier applied to Subiaco-Kings Park and Forrestdale resulted in much poorer accuracies (table 3.9). In Subiaco-Kings Park almost half the brown pixels were mislabelled (low producer's accuracy) whilst half the pixels labelled as shadow were not actually in shadow (low user's accuracy). In Forrestdale a low

brown producer's accuracy, low green vegetation user's accuracy and very low shadow user's accuracy resulted in a very low overall accuracy of 62%.

The results of the nDSM thresholded classifier applied to a wider region around Rockingham were visually assessed. The classification appeared to correctly label most vegetation, ground and roofs $25km$ both East and South of Rockingham (East to the Darling Ranges and South to Mandurah). Although small patches of poor behaviour were observed in dense forests and confusion with blue roofs and water occurred occasionally.

Just North of Rockingham the shadow classification was significantly degraded. This degradation did not appear to correlate to changes in soil type, building density, or landscape type.

To the West of Rockingham was ocean so this direction was not considered.

Discussion

These experiments showed that CVAR can be used to generate high quality multiclass classifiers that work over at least $16km^2$ regions. All of the classifiers extracted at least four spectral classes and combined with a threshold on the nDSM to create thematic maps with at least seven classes. The overall accuracies for these nDSM-separated classifications were above 75% for all three regions.

For simplicity these classifications were generated using only point data (except for the generation of the nDSM) however accuracies could improve significantly if contextual features were incorporated (many classifiers in the literature use spatial context [52, 102, 55]).

Both site-wise and directed CVAR comparisons were used in these experiments and neither method was clearly superior. A purely site-wise generated classifier and a purely directed-comparison generated classifier were both trained in the Forrestdale and Jandakot regions. These classifiers performed similarly in Forrestdale and Rockingham, whilst in Subiaco-Kings Park the directed classifier achieved a slightly better overall accuracy but was poorer at extracting brown cover.

One benefit of directed comparisons is that the separations were not dominated by illumination differences. Conversely the site-wise separations were often dominated by brightness wherein super-bright sites dominated the CVAR analysis causing most of the typical sites sites to cluster near the shadow and dark

Subiaco-Kings Park														
Subiaco-Kings Park trained	automatic	manually interpreted label assigned to pixel												Abbreviations
		bt	bn	cc	gv	gy	pls	snv	sv	sg	o	uk	sum	user's	bt = black tar
	bt	35	5	0	1	5	0	3	0	0	0	4	49	71.4	bn = brown
	bn	4	33	1	3	2	0	0	0	0	1	10	44	75	cc = concrete
	cc	1	0	3	0	4	0	0	0	0	0	5	8	37.5	gv = green vegetation
	gv	0	1	0	82	0	1	0	1	0	0	5	85	96.5	gy = grey
	gy	10	0	1	1	46	0	2	0	1	2	6	63	73	pls = pools/water
	pls	0	0	0	0	0	0	0	0	0	0	0	0	NaN	snv = shadowed non-veg
	snv	0	0	0	0	0	0	0	1	0	0	5	1	0	sv = shadowed veg
	sv	0	1	0	6	0	0	1	4	0	0	21	12	33.3	sg = synthetic green
	sg	0	0	0	0	0	0	0	0	0	0	1	0	NaN	o = other covers
	sum	50	40	5	93	57	1	6	6	1	3		262		uk = unknown
	producer's	70	82.5	60	88.2	80.7	0	0	66.7	0	0				

Table 3.1: Confusion matrix for the Subiaco-Kings Park trained classifier applied to the Subiaco-Kings Park region. The sums of each column and row, and user's and producer's accuracies (in percentages) are also shown. Overall accuracy was 77.5%. The Unknown values (shown in grey) were ignored for the accuracy statements.

Subiaco-Kings Park												
Subiaco-Kings Park trained	automatic	manually interpreted label										Abbreviations
		bg	br	gg	gt	GG	Gr	s	o	uk	sum	user's	bg = brown ground
	bg	10	0	1	0	5	0	0	0	4	16	62.5	br = brown roof
	br	2	14	0	0	0	1	0	0	13	17	82.4	gg = green grass
	gg	0	0	12	9	0	0	1	1	6	23	52.2	gt = green tree
	gt	0	0	3	45	0	0	0	0	12	48	93.8	GG = grey ground
	GG	2	0	1	1	35	0	3	2	13	44	79.5	Gr = grey roof
	Gr	1	1	0	0	10	41	4	0	25	57	71.9	s = shadow
	s	0	0	1	3	0	0	21	0	15	25	84	o = other
	o	0	0	0	0	0	0	0	0	1	0	NaN	uk = unknown
	sum	15	15	18	58	50	42	29	3		230		
	producer's	66.7	93.3	66.7	77.6	70	97.6	72.4	0				

Table 3.2: Confusion matrix for the Subiaco-Kings Park trained classifier with a 300mm nDSM threshold applied to the Subiaco-Kings Park region. Due to the rarity of pools and synthetic green materials they have been added to the *Other* class. White roofs, concrete roofs and above-the-ground black tar were combined to form the grey roof class due to the difficulty in manually discriminating between these types. For a similar reason the shadow classes have been combined. The overall accuracy was 77.4%. The Unknown values (shown in grey) were ignored for the accuracy statements.

Forrestdale

Jand. & Forr. - site-wise

automated	manual interpretation							
	bn	gv	gy	s	o	uk	sum	user's
bn	412	9	12	21	0	64	454	90.7
gv	25	131	0	30	0	42	186	70.4
gy	73	0	162	6	4	40	245	66.1
s	16	2	2	48	0	8	68	70.6
sum	526	142	176	105	4		953	
producer's	78.3	92.3	92	45.7	0			

Forrestdale

Jand. & Forr. - directed

automated	manual interpretation							
	bn	gv	gy	s	o	uk	sum	user's
bn	433	28	11	20	0	82	492	88
gv	3	86	0	5	0	23	94	91.5
gy	61	0	159	5	3	31	228	69.7
s	29	28	6	75	1	18	139	54
sum	526	142	176	105	4		953	
producer's	82.3	60.6	90.3	71.4	0			

Abbreviations

bn = brown
gv = green veg
gy = grey
s = shadow
o = other
uk = unknown

Table 3.3: Confusion matrices of the classifiers trained in Jandakot and Forrestdale, and applied to the Forrestdale region (spectral only classifiers). Overall accuracy was 79.0% for both directed and site-wise. The Unknown values (shown in grey) were ignored for the accuracy statements.

Forrestdale

Jand. & Forr. - site-wise

automated \ manual interpretation	bg	br	gg	gt	GG	Gr	s	o	uk	sum	user's
bg	384	0	1	2	11	0	18	2	61	418	91.9
br	21	0	1	4	1	0	3	2	7	32	0
gg	16	0	17	10	0	0	10	0	16	53	32.1
gt	1	0	9	95	0	0	20	6	28	131	72.5
GG	71	0	0	0	153	1	6	0	38	231	66.2
Gr	2	0	0	0	4	4	0	4	2	14	28.6
s	15	0	0	2	1	1	48	1	8	68	70.6
sum	510	0	28	113	170	6	105	15		947	
producer's	75.3	NaN	60.7	84.1	90	66.7	45.7	0			

Forrestdale

Jand. & Forr. - directed

automated \ manual interpretation	bg	br	gg	gt	GG	Gr	s	o	uk	sum	user's
bg	399	0	5	5	11	0	17	2	72	439	90.9
br	22	0	3	14	0	0	3	6	15	48	0
gg	2	0	12	4	0	0	0	0	7	18	66.7
gt	0	0	3	67	0	0	5	0	17	75	89.3
GG	60	0	0	0	149	1	5	0	29	215	69.3
Gr	1	0	0	0	5	4	0	3	2	13	30.8
s	26	0	5	23	5	1	75	4	18	139	54
sum	510	0	28	113	170	6	105	15		947	
producer's	78.2	NaN	42.9	59.3	87.6	66.7	71.4	0			

Abbreviations

bg = brown ground
br = brown roof
gg = green grass
gt = green tree
GG = grey ground
Gr = grey roof
s = shadow
o = other
uk = unknown

Table 3.4: Confusion matrices of the classifiers trained in Jandakot and Forrestdale, and applied to Forrestdale with a 300mm nDSM threshold. Overall accuracy was 74.0% and 74.6% for site-wise and directed respectively. Eleven of the the *other* pixels were brown trees or bushes, which were in the *brown* class in table 3.3. The Unknown values (shown in grey) were ignored for the accuracy statements.

ROCKINGHAM

Jand. & Forr. - site-wise

automated	manual interpretation							
	bn	gv	gy	s	o	uk	sum	user's
bn	25	8	5	6	0	14	44	56.8
gv	0	29	0	6	1	3	36	80.6
gy	6	0	36	1	1	3	44	81.8
s	0	0	0	0	1	0	1	0
sum	31	37	41	13	3		125	
producer's	80.6	78.4	87.8	0	0			

ROCKINGHAM

Jand. & Forr. - directed

automated	manual interpretation							
	bn	gv	gy	s	o	uk	sum	user's
bn	27	10	6	10	1	15	54	50
gv	0	27	0	1	0	3	28	96.4
gy	4	0	35	1	1	2	41	85.4
s	0	0	0	1	1	0	2	50
sum	31	37	41	13	3		125	
producer's	87.1	73	85.4	7.7	0			

Abbreviations
bn = brown
gv = green veg
gy = grey
s = shadow
o = other
uk = unknown

Table 3.5: Confusion matrices of the classifiers trained in Jandakot and Forrestdale and applied to Rockingham. Overall accuracies were 72.0% for both directed and site-wise. The Unknown values (shown in grey) were ignored for the accuracy statements.

SUBIACO-KINGS PARK

Jand. & Forr. - site-wise

automated	manual interpretation							
	bn	gv	gy	s	o	uk	sum	user's
bn	24	2	2	9	0	8	37	64.9
gv	6	88	3	12	2	13	111	79.3
gy	10	2	107	6	1	9	126	84.9
s	0	1	0	12	2	0	15	80
sum	40	93	112	39	5		289	
producer's	60	94.6	95.5	30.8	0			

SUBIACO-KINGS PARK

Jand. & Forr. - directed

automated	manual interpretation							
	bn	gv	gy	s	o	uk	sum	user's
bn	30	7	9	4	1	16	51	58.8
gv	2	83	1	6	1	5	93	89.2
gy	7	1	100	2	2	4	112	89.3
s	1	2	2	27	1	5	33	81.8
sum	40	93	112	39	5		289	
producer's	75	89.2	89.3	69.2				

Abbreviations
bn = brown
gv = green veg
gy = grey
s = shadow
o = other
uk = unknown

Table 3.6: Confusion matrices of the classifiers trained in Jandakot and Forrestdale, and applied to the Subiaco-Kings Park region. Overall accuracy was 80.0% and 83.0% for the site-wise and directed classifiers respectively. The Unknown values (shown in grey) were ignored for the accuracy statements.

Rockingham										
Rockingham trained	automated	manual interpretation								
		bn	gv	gy	s	bp	b	uk	sum	user's
	bn	29	6	5	2	0	0	14	42	69
	gv	0	29	0	2	0	0	5	31	93.5
	gy	2	0	35	1	1	1	1	40	87.5
	s	0	2	0	8	0	0	0	10	80
	bp	0	0	1	0	1	0	0	2	50
	b	0	0	0	0	0	0	0	0	NaN
	sum	31	37	41	13	2	1		125	
	producer's	93.5	78.4	85.4	61.5	50	0			

Abbreviations
bn = brown
gv = green veg
gy = grey
s = shadow
b = blue
bp = bright pools
uk = unknown

Table 3.7: Confusion matrix for the Rockingham-trained classifier applied to Rockingham. Overall accuracy was 81.6%. The Unknown values (shown in grey) were ignored for the accuracy statements.

Rockingham													
Rockingham trained	automated	manual interpretation											
		bg	br	gg	gt	GG	Gr	s	b	bp	uk	sum	user's
	bg	20	0	4	0	4	0	0	0	0	11	28	71.4
	br	1	8	0	2	1	0	2	0	0	3	14	57.1
	gg	0	0	16	0	0	0	1	0	0	3	17	94.1
	gt	0	0	0	13	0	0	1	0	0	2	14	92.9
	GG	2	0	0	0	20	0	0	0	0	0	22	90.9
	Gr	0	0	0	0	0	15	1	1	1	1	18	83.3
	s	0	0	0	2	0	0	8	0	0	0	10	80
	b	0	0	0	0	0	1	0	1	0	0	2	50
	bp	0	0	0	0	0	0	0	0	0	0	0	NaN
	sum	23	8	20	17	25	16	13	2	1		125	
	producer's	87	100	80	76.5	80	93.8	61.5	50	0			

Abbreviations
bg = brown ground
br = brown roof
gg = green grass
gt = green tree
GG = grey ground
Gr = grey roof
s = shadow
b = blue
bp = bright pools
uk = unknown

Table 3.8: Confusion matrix for the Rockingham-trained classifier with a $300mm$ nDSM threshold applied to Rockingham. Overall accuracy was 80.8%. The Unknown values (shown in grey) were ignored for the accuracy statements.

Subiaco-Kings Park

Rockingham trained: automated	manual interpretation									
	bn	gv	gy	s	b	bp	o	uk	sum	user's
bn	22	2	3	1	0	0	1	6	29	75.9
gv	3	75	1	2	0	0	1	6	82	91.5
gy	8	2	86	3	0	0	1	9	100	86
s	7	14	8	33	1	0	0	9	63	52.4
b	0	0	8	0	0	0	1	0	9	0
bp	0	0	6	0	0	0	0	0	6	0
o	0	0	0	0	0	0	0	0	0	NaN
sum	40	93	112	39	1	0	4		289	
producer's	55	80.6	76.8	84.6	0	NaN	0			

Forrestdale

Rockingham trained: automated	manual interpretation								
	bn	gv	gy	s	b	o	uk	sum	user's
bn	353	1	12	5	0	0	47	371	95.1
gv	11	85	0	4	0	0	23	100	85
gy	89	1	155	4	0	2	35	251	61.8
s	70	55	9	90	1	1	49	226	39.8
b	3	0	0	2	0	0	0	5	0
o	0	0	0	0	0	0	0	0	NaN
sum	526	142	176	105	1	3		953	
producer's	67.1	59.9	88.1	85.7	0	0			

Abbreviations

bn = brown
gy = grey
gv = green veg
s = shadow
b = blue
o = other
bp = bright pools
uk = unknown

Table 3.9: Confusion matrices for the Rockingham classifier applied to Subiaco-Kings Park and Forrestdale. Overall accuracies were 75% and 71.7% for Subiaco-Kings Park and Forrestdale respectively. The Unknown values (shown in grey) were ignored for the accuracy statements.

water sites. For example in both Forrestdale and Rockingham extremely bright sites had to be removed before the majority of each class could be extracted resulting in many more stages than in the directed classifiers.

Application of the classifiers to regions that they were not trained in showed that they did not generalise to larger regions very predictably. The Rockingham-trained classifier appeared to function well for $25km$ East and South of Rockingham. But visual inspection in other areas revealed that the accuracy of the Forrestdale and Jandakot trained classifiers varied every few image frames close to Forrestdale.

The experiments also showed that shadow detection was often the least reliable. In the Forrestdale region all the classifiers tested had very low shadow accuracy (either user's or producer's accuracies were below 55%). In Subiaco-Kings Park the directed Forrestdale-Jandakot trained classifier and the Subiaco-Kings Park trained classifier both extracted shadows with reasonable accuracy (when the latter's shadow classes were combined), but the other two classifiers had very low shadow accuracies. In the Rockingham region not enough shadows were sampled to reliably ascertain the accuracy of the classifiers.

The manual interpretations were relatively easy to implement on regular grids, however regular sampling suffers from spatial correlation [30]. Visual inspection was also required to check the accuracy of the rarer cover types that this scheme failed to sufficiently sample. The accuracies calculated for these experiments were thus not very precise.

3.1.3 Comparison to CVA and CVAR with Independent Denominators

Here we compare CVA, CVAR, and a more flexible version of CVAR. This more flexible version of CVAR with independent denominators (CVARid) has a different denominator $a + c \cdot x$ for each band so that the simplex search is sixteen dimensional instead of four.

Technically $a \neq 0$ means that the CVA transform is always contained in the CVAR search space and given a perfect search algorithm the canonical root of a CVAR analysis should always be greater than the CVA canonical root. Similarly the canonical roots from CVARid should always be greater than standard CVAR.

Separations given by all three analyses were compared for different groups from the Subiaco-Kings Park training data (from experiments in section 3.1.2).

The canonical root was not always a good indication of the quality of the separation so the training pixels were also plotted against the first two canonical transforms. The primary issue observed was groups that appeared scattered and sparse on the plots and thus had a large variance of transformed value. A scattered group suggested that either the grouping was incorrect (the group contained pixels that were too different) or that a singularity of the transform (for CVAR and CVARid) was close to the spectral values of the pixels (the denominator values were close to 0). The more scattered a group the more unpredictable the transform would be when applied to the full images because very similar pixels could have large differences in transformed value, especially if the denominator changed sign (eg. a homogenous roof could be split into very large positive and negatie values).

Figures 3.5 and 3.6 are example plots of this comparison showing the flaws that are possible even with high canonical roots. CVARid achieved the highest canonical root in both of these examples however the groups experienced extreme scattering/stretching and would not be very useful for classifying the full image.

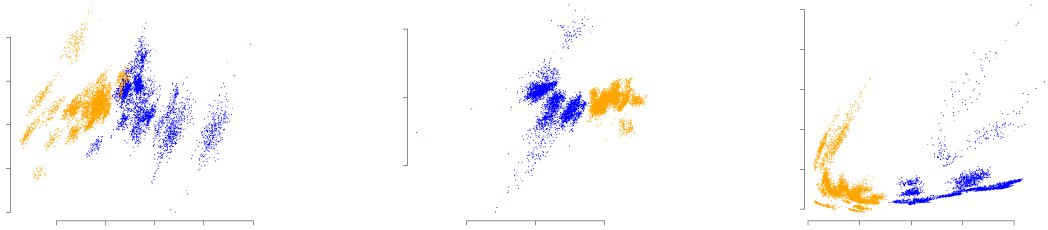


Figure 3.5: Results of CVA (left), CVAR (centre) and CVARid (right) of a direct contrast between brown training pixels (in orange) and grey training pixels (in blue). The canonical roots were 1.4, 4.5 and 5.1 respectively. The pixels have been transformed by the first and second canonical transforms found in each process. The two groups were mixed after the CVA transforms but not after the CVAR or CVARid transforms. Both the CVAR and CVARid transformations stretched the groups, although the effect was much larger from the CVARid transforms. This stretching makes it difficult to choose discrimination boundaries.

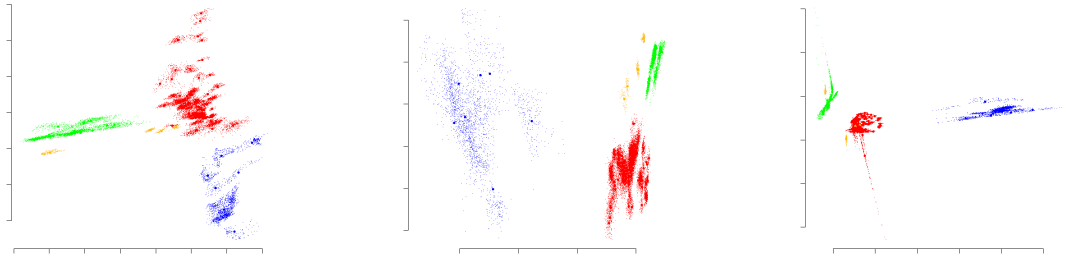


Figure 3.6: Results of CVA (left), CVAR (centre) and CVARid (right) of a directed contrast between four groups: pools (in blue), synthetic green roofs (in gold), green vegetation (in green) and all the other training pixels (in red). The canonical roots were 8.2, 18.7 and 30.9 respectively. The pixels have been transformed to the first and second canonical transforms found in each process. Here the CVAR transforms clearly out-performed CVA and CVARid; the groups were further apart than CVA and were fairly contained (not scattered) compared to the CVARid transformations.

Of the sixteen different directed contrasts tested, CVAR performed at least as well as CVA in all but one case. The exception occurred with a two-group separation between black tar and concrete. The simplex search must have converged to a local minimum, in hindsight the result may have been improved by choosing an initial simplex within the bound given in section 3.1.5.

In five of the cases CVAR was able to separate classes when CVA could not, with only occasional extreme stretching and skewing of the pixel values.

CVARid found bigger canonical roots than CVAR in all of the cases, however it usually created more extreme separations with a greater scattering of pixels.

In these quick experiments CVAR almost always outperformed CVA and CVARid for generating classification indices and enabling decision boundaries.

More comprehensive experiments on other data including comparisons of site-wise separations are required to determine the usefulness of CVARid.

3.1.4 CVAR Topographic Illumination Normalisation

An example region transformed by the CVAR canonical transforms and CVA canonical vectors trained on the same data is shown in figure 3.7. Note that

CVAR was not perfect; not all roofs have been perfectly normalised.

So long as the training data contains a good spread of cover types and illuminations then the CVAR canonical transforms should reduce illumination differences. The exception is when the separation is dominated by brightness as was the case in the site-wise separations in section 3.1.2.

3.1.5 Brute Force Search of Starting Simplex

During the pilot studies it was observed that occasionally repeat CVAR analyses would not produce exactly the same result and in section 3.1.3 a local minimum was observed in one of the comparisons. This suggested that the size and shape of the initial simplex influenced the quality of the final result.

It was theorised that part of the difficulty was caused by transformed points approaching infinity during the simplex search. This could have been caused by initial c_2 guesses such that the denominator of CVAR was close to zero ($a+c_2 \cdot x \approx 0$) for a few training pixels. The magnitudes of the transformed pixel values would then be very sensitive to changes in c_2 causing the simplex search to be very sensitive to the changes in the starting simplex.

The size of $|c_2 \cdot x|$ ranges from 0 to $||c_2|| \cdot \max\{||x|| : x \in \text{training data}\}$ so a c_2 such that

$$||c_2|| < \frac{1}{\max\{||x|| : x \in \text{training data}\}}$$

will keep $|c_2 \cdot x| < 1$ and the denominator away from 0.

Thus starting simplex vertices such that each ordinance is smaller than $\frac{1}{\max\{||x||:x\in\text{training data}\}}$ should start the search away from sharp instabilities and consequently yield more reliable results.

Nine different CVAR comparisons from the pilot studies were chosen to experimentally investigate the influence of simplex shape and size. These included site-wise, group-wise, and directed two-group comparisons.

In the experiments the size of the starting simplex was varied from 10 to 10^{-5}, each size was repeated five times. The vertices for each initial simplex were chosen randomly between $[-1, 1]$ with each vertex then scaled by the size factor.

For seven of the nine training sets start simplices with sizes smaller than $\frac{1}{\max\{||x||:x\in\text{training data}\}}$ correlated with the simplex search converging to the same canonical root nearly all of the time. Usually this canonical root was the best or

Figure 3.7: CVAR canonical variables normalise terrain illumination. CVAR and CVA were applied to the same set of training data with the same groups. The first two CVA canonical variables *(left)* and the first two CVAR canonical transforms *(right)* are shown for a small excerpt of multispectral data *(top - shown in true colour)*. The first and second canonical transforms are displayed in red and green respectively. Due to the terrain illumination effect the South-West faces of the roofs appear darker than the North-East faces in the multispectral data. The CVAR canonical transforms have normalised this effect in each of the brown, white and orange roofs (the values are constant for each roof). In comparison the light and dark sides of the roofs are still visible in the CVA canonical variables. Note that the blue and black roofs were not normalised by the CVAR transforms; due to the distribution of the training data the CVAR separation has optimised the differences between the brown and white pixels.

very close to the best found, however there was one set of training data where much higher canonical roots were occasionally reached with larger starting simplices.

For this particular training set the separation found from small starting simplices allowed for the discrimination of some bright swimming pools and brown cover. Whilst the separations with much higher canonical roots were much poorer for classifications because they transformed a single swimming pool site a large distance away, causing the rest of the training data to clump together.

There were two cases where searches with starting simplices below the size bound found multiple canonical roots. In both cases the larger canonical roots corresponded to separations that were poorer for classifying due to a greater scattering of pixels.

When multiple simplex searches converge to differing canonical roots it is because either the simplex searches are converging to local minima or they are finishing prematurely (such as in a very flat search space). The presence of compact extreme (very distinct) groups could explain the former case because they create local minima: A c_2 such that $1 + c_2 \cdot x$ is close to 0 for an extreme group and no other groups can produce very high between-group variance (the extreme group can be very far away from the rest of the training data so the group means $\bar{x}_k$ are all further away from the total mean $\bar{x}_T$) whilst the within-group variance is not necessarily high because only a single group is scattered by the near-zero denominator.

The presence of scattered groups in the final separations supports this hypothesis (although it is possible for pixels to be scattered without the presence of extreme groups, for an example see figure 3.5).

Furthermore if the denominator can have a different sign in the extreme group to the other groups then very high canonical roots could easily be created: Moving such a denominator $1 + c_2 \cdot x$ closer to 0 for the isolated group both *increases* between-group variance (the extreme group is sent much further from the rest of the data) and *decreases* within-group variance (the denominator *increases* for all the other groups causing the data points to move closer to their respective group means $\bar{x}_k$).

Further research is needed to confirm this hypothesis and establish good methods for preventing premature finishing. The case where the denominator is a dif-

ferent sign for the extreme group would be particularly easy to investigate, and detect or prevent.

This experiment showed that the bound $\frac{1}{\max\{||x||:x\in\text{training data}\}}$ was a very good rule of thumb for choosing initial starting simplices. For all but two training sets CVAR that was initialised with a starting simplex smaller than this bound reliably produced the best separations for classification. The cases where the simplex search did not consistently converge to the same canonical root appeared to coincide with the creation of scattered groups, suggesting the presence of local minima caused by extreme groups. Thus training sets with extreme groups removed may yield more consistent simplex search results.

These results also suggest that the search spaces contain some structure that could be exploited. For example if the search did not allow opposite signs in the denominator then this could result in a CVAR that more reliably arrived at the most usable separations.

3.1.6 Guidelines to the Application of CVAR

Our CVAR implementation requires three inputs, a set of training data, a scale to set the size of the starting simplex, a tolerance as a finish criterion for the simplex search and a choice of a. Many other parameters in the algorithm (e.g. parameters of the simplex search, and removal of low variance directions) have been fixed (see appendix D.1 for more details).

As discussed in the introduction, any choice of $a \neq 0$ is equivalent. For ease set $a = 1$.

As shown in section 3.1.5 it is best to choose a starting simplex with vertices smaller than $\frac{a}{\max\{||x||:x\in\text{training data}\}}$. Within this bound the precise location of vertices does not have much impact and can be chosen randomly.

No investigation into the choice of tolerance was conducted. However the tolerances used in the experiments were always 1×10^{-4} or smaller and appeared to function quite well.

Without more knowledge of the search space it is best to repeat the CVAR analysis multiple times, if needed trying different tolerances and initial sizes.

Training Sites

CVAR acts upon a set of training data, in our application this is a set of manually interpreted pixels. Typically the training data was composed of many small *sites* containing a set of roughly homogeneous pixels. Each site should have a unique name that includes a description of the cover type it represents.

The more representative the training set is of the spectral variety of the region the more useful the results of the CVAR analysis will be. For example it is a good idea to train on pitched roofs using two sites, a site on the dark side and another on the bright side, to equalise effects of brightness and encourage the normalisation of the topographic illumination effect.

The pilot studies showed that shadow classification can be very good, however it was more commonly the case that shadows would be confused with many other classes and caused brightness to dominate the CVAR separation making it difficult to separate other cover types. In many applications it is expected that the shadow sites will need to be removed from the training data to achieve the best CVAR results.

The presence of data offsets (discussed in the NDVI section - section 3.2), where the ambient light is not removed or there is some other additive term, should be checked and removed. Crippen [32] advised that this check should occur for any method that involves band ratios.

Extreme pixels in the training data lead to poorer CVA vectors and should be avoided: CVA can be thought of as two transforms with the first transform using the direction of the largest within-groups variation and the second finding the maximum variation between the group means [19]. In the presence of extreme pixels the direction of the maximum within-group variation no longer represents the variation of the normal pixels. As a result the groups composed of the more normal pixels are not as compact as they could be. Due to the non-linear properties of the CVAR transforms the effects of these extreme values is probably much larger than with plain CVA.

Care should also be taken to use approximately the same sized training sites (or groups) because large groups can skew the CVA analysis. Too many similar sites could also skew the analysis, although this was not noticed in any of the investigations in this thesis.

Groups

The groups to use for the training data are not always obvious.

Heuristically the most robust option is *site-wise analysis* where each site is treated as an individual group. The separations expound the similarities and differences between sites. Thus a site-wise analysis can be an exploratory tool, revealing unexpected distinctions and similarities between cover types. It also reveals extreme pixels/sites, errors in the training data such as mislabelled sites and sites with large variation, and can also indicate spectral values that are not included in the training data.

When separations are not occurring naturally from site-wise comparisons, or the separation is messy (such as when it is dominated by illumination differences), then *directed contrasts* may be more fruitful. In a directed contrast the training sites are manually collated into two or more groups for the CVAR analysis.

For similar reasons to extreme pixels, extreme groups also lead to poorer results in multi-group analyses (unless one is particularly interested in discriminating that extreme group). The extreme group dominates the direction of maximum variance between group means and as a result the final separations between the other groups is poorer.

Results have not been conclusive, but it is expected that site-wise derived indices function more reliably over greater areas. This is because a directed analysis returns canonical vectors that are *forced* to reflect the variation between groups, and do not necessarily reflect the general variation of the training data. The canonical vectors from the directed analysis thus cause more distortion of the spectral space.

Application of CVAR canonical transforms

The CVAR canonical vectors may be used in very much the same way that CVA is used. The unique ability of CVAR is that the transforms can often reduce terrain illumination differences.

Often the first step is to plot the training data transformed by the canonical transforms. There are functions available within the implementation of CVAR for quickly generating these plots (see appendix D.1.1).

This plot can be used to explore the training data for clusters, look for omitted

cover types, generate indices and specify discrimination functions.

The canonical transforms or the indices manually created using the plot can be used directly (e.g. [7, 66, 103]) or form a feature in a classification scheme.

3.1.7 Discussion

CVAR is a supervised method for producing pixel-wise features suitable for discrimination. The features generated were typically robust to illumination differences, and were more useful than standard CVA features for separating classes. These features can be used to generate thematic maps using standard classification techniques; a hierarchical method based on manually determined boundaries was experimented with here.

The highly supervised hierarchical classifications produced multi-class thematic maps with greater than 75% accuracies for $5km^2 - 16km^2$ areas however the classifiers did not reliably generalise to larger areas implying many stratifications would be required to classify the entire Urban Monitor area. Due to time constraints this was beyond the scope of this research. The CVAR features were often unreliable in shadow and water due to the greater proportion of noise in low spectral values, it was suspected that the presence of these cover types in the training data also made it difficult for CVAR to extract canonical transforms for separating other cover types. Full classifications may eventually require shadow masks and water masks that are generated independently of CVAR.

The feasibility of CVAR has only been investigated here and in [28], both showing very promising results. Although the accuracies seen here are not as high as typically seen in the literature, the regions are larger and have greater diversity. In the future CVAR could be applied to a wide range of image data and problems, and possibly even general data mining problems. Experimentation with CVAR (and applying the guidelines in section 3.1.6) on the 2009 Urban Monitor data is an obvious step to take. It is expected that the greater consistency of the calibration in the 2009 data will enable more accurate multiclass classifications over larger areas. Another promising application is the possibility of generating a vegetation index that is much better than NDVI.

The impact of negative denominators in CVAR is an intriguing direction of exploration. In section 3.1.5 it was shown that initial simplices with positive

denominators for all pixels were likely to give the best separation and it was hypothesised that extreme separations could only occur when the denominator was such that it changed sign for just one group. This idea reappeared independently in the next section where it was noted that the denominator in NDVI, used for normalising the topographic illumination effect, should stay positive.

If the denominator should stay positive in CVAR then this suggests a version where $a = 0$ because then a c_2 restricted to positive coordinates yields positive denominators for any pixels with positive spectral values. Due to the scale invariant nature of CVA the magnitude of c_2 can be restricted to $\|c_2\| = 1$, effectively reducing the dimension of the search space. Furthermore if shadows and other dark pixels are avoided then divisions by very small values will only be possible when some components of c_2 approach zero and the orthogonal components of pixels are close to zero. Thus in this regime CVAR would still have the potential to stretch groups and create extreme separations, but due to the forced-positive denominator, extreme separations may not be as favoured by the simplex search.

Another benefit of this $a = 0$ option is that the denominator is then a linear combination of the spectral values. Thus the CVAR transforms (now $\frac{c_1 \cdot x}{c_2 \cdot x}$) match the ratioing theory and will theoretically produce consistent values for each material captured. When $a \neq 0$ consistent values were not supported by theory because the denominator could not be constructed by linear combinations of the spectral values.

Greater knowledge of the search space structure could provide a faster, more reliable search. Furthermore, the quality of the simplex search could be improved by investigations into optimal parameters. For example the best choice of tolerance has yet to be investigated.

It is evident from sections 3.1.3 and 3.1.5 that the search space contains some structure and an in depth mathematical analysis of the space could suggest a more appropriate search algorithm or better parameter choices.

3.2 Vegetation Classification using NDVI

For many environmental questions vegetation classes are the main cover type of interest (e.g. those relating to vegetation height, tree canopy and vigour). These vegetation classes also provide good prototypes to begin investigating the statis-

tics of decimetre resolution land cover maps. We used the Normalised Digital Vegetation Index (NDVI) to generate green vegetation masks of the $9600km^2$ Urban Monitor region with 94% accuracy for the 2007 data and 90% accuracy for 2009 data.

NDVI can be related to the physical properties of chlorophyll, the molecule that performs photosynthesis in plants. Chlorophyll is strongly absorbing in red light, but highly reflective in near infrared light [7] so the difference between near infrared reflectance and red reflectance $nir - r$ gives an indication of which pixels represent vegetation. This, combined with a denominator designed to eliminate illumination and atmospheric differences, leads to NDVI [78]

$$NDVI = \frac{nir - r}{nir + r}.$$

NDVI has been widely used in multispectral remote sensing of vegetation [101] with many applications to the monitoring of vegetation, and classification of urban areas (e.g. [23, 65]). Recently [10] used NDVI to generate a vegetation/non-vegetation classification at 0.5m GSD for the entire area of Italy.

The negative spectral values in the Urban Monitor data pose a peculiar problem for the application of NDVI. The negatives are caused by the empirical nature of the calibration, where the spectral values are calibrated to the reflectance of horizontal, matte, non-shadowed surfaces [29].

When NDVI is applied to a pixel with a large *negative* near-infrared value and a small *positive* red value it leads to a negative denominator and a *positive* NDVI. But in reality this pixel has reflected much more red light than infrared so the NDVI should be highly negative to indicate a non-vegetated pixel. The issue is that the numerator $nir - r$ represents both a direction (positive = more chlorophyll), and a magnitude. However a negative denominator switches the direction so that positive NDVI does not necessarily equate to more chlorophyll making NDVI difficult to use. Ideally the denominator would only affect the magnitude. Here we have chosen to truncate any negative values to the band's *origin* which is introduced in the next paragraph.

The presence of negative spectral values begs the question of whether zero really does correspond to an area from which the camera received no radiation. We refer to the value that represents no radiation leaving the ground as the *origin*. It is important because the theory behind ratioing [47] assumes radiance

is a combination of multiplicative factors with an insignificant additive ambient light term. Thus if no light is reflected from a surface then the recorded spectral value should be zero. If it isn't, and there is some offset due to camera calibration or processing, then the effect of angles (the sun illumination direction, the surface normal direction and the camera location) is not cancelled out by a ratio of the bands.

This is the *data offsets* problem that Crippen [32] demonstrated is essential to consider in any ratioing process. Crippen found that the most significant improvement to discrimination using ratio-based indices involved the translation of band values to a correct origin. The multispectral BRDF calibration [29] partially performed this data offset correction, however the results here showed that it was not always perfect. In the future the calibration technique will improve and it is likely that no further translation of band values will be required.

In searching through the 2007 mask it was found that zero represented shadows well in both nir and r bands. However in some large dense forests different origins were required.

In 2009 the multispectral images were not perfectly calibrated to ground reflectance leading to non-zero origins. However the offset was more consistent over the entire area and the same parameters worked everywhere, including large forests.

Apart from the negative values and the offset of spectral values, the sensitivity of fractions to low values meant a noise tolerance was also required.

The formula for calculating the NDVI was thus

$$NDVI = \begin{cases} NULL & \{nir < O_{nir} + tol_{nir}\} \cap \{r < O_r + tol_r\} \\ 1 & \{nir \geq O_{nir} + tol_{nir}\} \cap \{r < O_r\} \\ -1 & \{nir < O_{nir}\} \cap \{r \geq O_r + tol_r\} \\ \frac{(nir - O_{nir}) - (r - O_r)}{(nir - O_{nir}) + (r - O_r)} & elsewhere \end{cases} .$$

where O_{nir} and O_r are the origins for the nir and r bands respectively and tol_{nir} and tol_r are the tolerances for the nir and r bands respectively. Thus NDVI ranged between -1 and $+1$.

3.2.1 Methods

The origins O_{nir}, O_r, tolerances tol_{nir}, tol_r and a threshold for NDVI must be chosen to form a classifier.

In 2007 with $O_{nir} = 0, O_r = 0$ it was found that a threshold of 0.2 extracted most green vegetation but in some areas many roofs had an NDVI just within the threshold, and as expected it did not capture dry brown grass.

To remove the roofs a morphological segmentation algorithm was used to create a conservative estimate of roof pixels. Inflows (section 2.2) was applied to the nDSM with a height threshold of 1m, slope threshold of 25^0 and an area threshold of 1000 pixels. Segments with 50% or more of the boundary flowing outwards were kept as roofs. Anything with NDVI between 0.2 and 0.3, and in the roof mask was *not* labelled vegetation, anything above 0.3, or any non-roof pixel with NDVI above 0.2 was labelled vegetation. Roof pixels with NDVI greater than 0.3 were still labelled vegetation because it was noticed that the roof mask included a significant portion of green trees, especially the peaks of trees, whilst the only roofs with NDVI higher than 0.3 were blue (a blue cover rule was not implemented due to time restrictions and the rareness of blue pixels.)

The 0.2 threshold and the roof correction threshold of 0.3 were used for all the classifications because choosing the origin and tolerances was a more objective, repeatable method that provided sufficiently accurate results.

2007

The parameters for the 2007 vegetation mask were chosen in an ad-hoc manner compared to the parameters for the 2009 data. Applying NDVI with the standard origins $O_{nir} = 0$, $O_r = 0$ without any tolerances appeared to produced very good vegetation masks over many areas (ignoring errors in shadows because tolerances had not been selected). A more detailed investigation into origins whilst searching for appropriate parameters within a large forest in the South (Myalup State Forest) confirmed that origins of $O_{nir} = 0$, $O_r = 0$ worked well *outside* the forest. Investigation of the tolerances showed that $tol_{nir} = 500$ and $tol_r = 500$ gave the best results over the whole Urban Monitor region. Smaller tolerances labelled too much shadow as vegetation, whilst larger tolerances removed too much true vegetation.

Investigation of parameters inside the Myalup State Forest found that origins of $O_{nir} = 300$ and $O_r = -300$, and tolerances of $tol_{nir} = 200$ and $tol_r = 200$, extracted the plantation trees and performed well in patches of native vegetation within the forest.

As noted in chapter 4 there were two other areas where this vegetation mask failed to work properly, a pine plantation in the North, and the forest in the South of the Darling Ranges.

In the Northern forest vegetation at ground level was removed using the nDSM because it was known that no low lying vegetation grows in these plantations. Using different origins and tolerances did not work because a large amount of cloud shadow caused many large changes to the spectral values in the shadows cast by the trees (the calibration didn't detect or account for cloud shadows).

The forest in the Darling Ranges was natural and contained significant amounts of ground level vegetation. In a number of locations and for each band in isolation, various thresholds that masked very dark/shadowed pixels, ranging from conservative masks to generous masks were recorded. The origins were chosen to lie within these ranges and the tolerances were then chosen to detect all of the shadows (this meant some non-shadow was below the tolerance in either band, but rarely both bands). The final parameters were $O_{nir} = 300$, $O_r = -300$, $tol_{nir} = 200$, $tol_r = 200$, which surprisingly were the same parameters found for the Myalup State Forest.

2009

To choose the origins and tolerances for the entire 2009 Urban Monitor data the same methods as in the 2007 Darling Ranges were used. Small areas in the South, North, urban regions, hills, farmland and forests were all checked. The final parameters were $O_{nir} = -50$, $O_r = 75$, $tol_{nir} = 100$, $tol_r = 100$. The resulting vegetation mask was double checked over multiple areas.

The same parameters functioned in dense forests and other areas suggesting that the 2009 calibration was much more spatially consistent than the 2007 calibration. The non-zero origins suggest that the data was not quite calibrated to ground reflectance.

3.2.2 Results and Discussion

Comparisons with the manually interpreted grids in the three pilots of section 3.1.2 were used to estimate the accuracy of the vegetation masks. There were much fewer Unknown points in these confusion matrices (table 3.10) compared to the multiclass confusion matrices (tables 3.1 - 3.9) because it was much easier to distinguish whether points were green-vegetation than to discriminate between narrower (and spectrally more similar) cover types such as road, concrete, brown pavement, sand or dirt.

In these comparisons shadows were treated as an illumination effect rather than a type of cover (shadows were considered a separate class in the multiclass classifiers of section 3.1). For example any points manually interpreted as a tree in shadow were assigned to the tree class, similarly any green grass in shadow was assigned the green vegetation class and any shadowed non-vegetated pixels were assigned the non-green-vegetation class. Shadowed points were assigned the unknown label if the underlying cover type could not be discerned.

The 2007 mask achieved an overall commission error of 1.2% and omission error of 3.4% (see table 3.10). These manual interpretations were updated for the 2009 data and used to analyse the accuracy of the 2009 vegetation mask (also in table 3.10). The 2009 accuracy was slightly poorer with more commission and omission errors.

The accuracy of a classifier that labelled any vegetation above $2m$ as tree canopy was also calculated (see table 3.11). In 2007 it detected trees with a commission error of 0.5% and omission error of 7.0%. In 2009 the commission error rate was similar and the omission error was slightly higher resulting in a slightly lower overall accuracy. In both years the tree mask included less than 60% of all the tree points. Most of these omissions were due to shadow (52 of the 2009 omission errors), DSM errors (at least 29 of the 2009 omission errors) and errors in the manual interpretation (at least 20 of the 2009 omission errors).

The manually interpreted grids were on the coastal plain in urban/peri-urban areas. The accuracy in the far North and South, the Darling Ranges and large forests is likely to be slightly different and more comparisons are required to get a full picture of the accuracy.

The quality over the full area was investigated by visual comparison of the

2007 Vegetation Mask					
automated	manual				
	not gv	gv	uk	sum	users
not gv	1087	70	123	1157	93.9
gv	17	274	37	291	94.2
sum	1104	344		1448	
producer's	98.5	79.7			

2009 Vegetation Mask					
automated	manual				
	not gv	gv	uk	sum	user's
not gv	994	99	147	1093	90.9
gv	39	317	12	356	89
sum	1033	416		1449	
producer's	96.2	76.2			

Abbreviations	
not gv =	non-vegetation, brown (dry) grass and brown bushes
gv =	green vegetation and brown trees
uk =	unknown

Table 3.10: Confusion matrices for the 2007 and 2009 NDVI vegetation classifications. *For 2007:* Commission Error = 1.2%, Omission Error = 4.8%, Overall accuracy = 94.0%. *For 2009:* Commission Error = 2.7%, Omission Error = 6.8%, Overall accuracy = 90.5%. The Unknown values (shown in grey) were ignored for the accuracy statements.

2007 $2m+$ Tree Mask					
automated	manual				
	not tr	tr	uk	sum	users
not tr	1219	112	106	1331	91.6
tr	8	147	16	155	94.8
sum	1227	259		1486	
producer's	99.3	56.8			

2009 $2m+$ Tree Mask					
automated	manual				
	not tr	tr	uk	sum	users
not tr	1160	134	142	1294	89.6
tr	5	162	5	167	97
sum	1165	296		1461	
producer's	99.6	54.7			

Abbreviations	
not tr =	non-vegetation, grass and bushes
tr =	$2m+$ trees (green and brown)
uk =	unknown

Table 3.11: Confusion matrices for the 2007 and 2009 tree classifications using the vegetation mask and the nDSM with a $2m$ threshold. *For 2007:* Commission Error = 0.5%, Omission Error = 7.5%, Overall accuracy = 92.0%. *For 2009:* Commission Error = 0.3%, Omission Error = 9.2%, Overall accuracy = 90.5%. The Unknown values (shown in grey) were ignored for the accuracy statements.

aerial photographs and the vegetation mask (greater detail of the methods and results can be found in chapter 4). The 2007 mask contained some roofs in some areas (especially blue roofs) and some particularly brown trees were omitted. The 2009 mask was not checked as thoroughly, but appeared to contain much fewer roofs. Blue roofs typically had a very high NDVI and were usually labelled vegetation. The removal of dark pixels through the origin and tolerances successfully avoided most water and shadows (no shadow and only a few water commission errors found - see chapter 4).

3.3 Conclusion and Future Work

This chapter investigated two different classification techniques. The first, using CVAR, was a training-based method capable of generating indices for discriminating between any spectrally different classes. The other classifier thresholded a single model based index, NDVI, and merged it with a roof mask to create a mask of photosynthetically-active (green) vegetation.

The NDVI classifier was created when it was found that, with the current methods, CVAR-trained classifiers did not generalise to large or predictable regions; CVAR classifiers covering the full Urban Monitor region would have required more strata than time allowed for training. At the very least a vegetation/non-vegetation classifier was required to begin considering the difficulties of environmental indicators.

The green vegetation classifier functioned with 94% and 90% accuracy over a few test areas for 2007 and 2009 respectively and extensive visual investigation showed similar quality across the whole Urban Monitor region. The NDVI classifier could not detect brown vegetation, however in the future brown trees could be extracted using morphological or textural features. The NDVI-based classifier also consistently commission blue roofs as green vegetation. This was deemed too time consuming to fix during development, however it should be possible to correct in the future using other spectral based features. The use of NDVI proved educational for the design of CVAR, showing the need to consider positive-only denominators, data offsets and removal of dark pixels.

CVAR was found to consistently outperform CVA and was able to produce features that significantly reduced the topographic illumination effects. The CVAR

classifiers were able to function over much larger areas than other classifiers seen in the literature. This was in part thanks to the quality of the BRDF calibrations performed on the spectral data [29]. In one study a classifier appeared to function well over a $625km^2$ area. In another experiment the calibrated spectral data over a small region of Perth was so consistent that CVAR was able to discern concrete from grey/white roofs, and partially separate shadowed vegetation from other shadows. However in other areas CVAR had difficulty when the training data included cloud shadows, dark shadows and water due to the noise inherent in this data and the large illumination difference to well lit surfaces.

It is anticipated that improved methods (see the guidelines in section 3.1.6) will allow multiclass classifications over the full region. Due to time constraints the improvements gained from these changes have not been tested.

The data offset problem was found to be a significant factor in the NDVI classifier and was probably a significant factor in the CVAR classifiers as well. Both could be improved using more robust and heuristically justified methods (suggested in [32]) to choose the origins of each band.

Chapter 4

Application to Full Urban Monitor Region

The application of the GEM algorithm (chapter 2) and the NDVI-based classifier (section 3.2) to the full $9600km^2$ Urban Monitor region was a non-trivial task due to the volume of data. This chapter presents the techniques used to accomplish this task, generate a few other maps over the whole region, manually inspect the data and correct any gross errors. Due to time restrictions no multidate comparison maps were created.

The manual inspection and correction of gross errors occurred using various specifically designed tools. Most errors that warranted repair were GEM issues in forested areas. The most commonly noticed vegetation classification errors were omission of brown vegetation and commission of blue roofs.

This data is already being used by urban decision makers (e.g. [100, p13]) and others (e.g. [97]) have demonstrated its usefulness to their particular domain. As an example application the total tree and green vegetation area for the full $9600km^2$ region were calculated for each year, along with the mean and standard deviation of tree canopy height (section 4.2.1).

The final data occupied approximately 20 Terabytes of storage.

The data sets that were generated are listed in table 4.1.

The methods for both generation of the data sets, and the manual inspection and correction are detailed in section 4.1. In section 4.2 the results are summarised and the methods discussed.

4.1 Methods

Each year collection occurred in a piecewise manner with the aircraft taking multiple weeks to photograph the entire region. This has occurred each year between January and March since 2007 (see [16] for more information on the data collection)[1]. For this project only the 2007 and 2009 data was used due to time constraints.

The ground mask, GEMs and intermediate roof masks were generated from the DSMs using the high performance computing facility, iVEC.

The data sets were split into 73 tiles of approximately $13km \times 15km$ and the remaining maps generated on local computers (tiles shown in [13, section 4])[2].

Manual checking for large, gross and repairable errors then occurred requiring approximately four hundred people hours.

Throughout all the management of data, version control and records of manual corrections were essential. This occurred through a strong file naming convention and the generation of a history file associated with each data file. The history documents were designed to be *independent* of the file name conventions so that if a file name did not follow the convention, the processing performed on the data and any derivative data sets could still be tracked through the history files.

A detailed description of the iVEC processing is given in section 4.1.1. The processing performed on local computers is described in section 4.1.2 and the manual checking and correction of products is described in section 4.1.3.

As an example of the power of this data the total vegetation and tree areas in the entire region for both years have been calculated (section 4.2.1).

4.1.1 Generating the Candidate Ground Points, GEM and Roof Mask

The candidate ground points and the GEM were created from the DSM using the programs from chapter 2. Due to memory restrictions the GEM was generated on pieces of 20000 by 20000 pixels overlapping by 1000 pixels, and then feathered

[1] the collection of the data and processing to produce the DSMs and calibrated orthophotos was the work of others

[2] the splitting into tiles and execution on iVEC was managed by colleagues Drew Devereux and Xiaoliang Wu

together. The two different parameter sets for fitting a GEM in the hills or the plains (see section 2.4) were chosen according to whether the mean height of the piece was above $100m$. The precise location of the boundary was not very important because at the foot of the Darling Ranges the results from either parameter set were very similar due to fewer buildings, smoother valleys and smoother ridges.

A roof mask was then created from a temporarily generated nDSM according to the methods in section 3.2. For this project the value of the nDSM was defined as

$$\text{nDSM} = \max(\text{DSM} - \text{GEM}, 0).$$

Division of these data sets into the seventy-three tiles then occurred.

This processing was performed by Drew Devereux and Xiaoliang Wu.

4.1.2 Generating the Remaining Maps

The remaining maps were generated on standard desktops in the following order

1. nDSM
2. vegetation mask
3. vegetation heights and vegetation index
4. green grass mask and bushes/trees mask
5. no data mask.

The nDSMs were created from the GEM and DSM with a simple differencing program.[3] The vegetation masks were created using the roof mask and NDVI according to parameters in section 3.2. The vegetation height maps were simply the intersection of the vegetation mask with the nDSM to yield the height of vegetation above the ground in millimetres.

The vegetation (vigour) index was calculated using the standard $NDVI = (nir - r)/(nir + r)$ for every pixel in the vegetation mask. Vegetation pixels with negative values of nir were assigned an index of -1, pixels with negative r values were assigned an index of $+1$, and the vegetation masks did not allow both nir

[3] In the future there is no reason why the nDSMs generated on iVEC could not be used.

and r to be negative so this case wasn't considered. No data offsets were used for these calculations.

The green grass mask and bush/trees mask were both created by thresholding the vegetation heights map at 500mm.

The last map, a no-data mask, was generated by searching for null values in the nDSM and for pixels in the multispectral images with *all four* bands null.

Detailed instructions for the execution of the programs used can be found in appendix E.2, the programs used are detailed in table E.3.

4.1.3 Distributed Manual Inspection and Correction of Products

The checking of all seventy-three tiles for both years of data was conducted over about four weeks using approximately four hundred people hours. There was a significant increase in speed once both dates of data became available as many gross errors were highlighted as gross change between the corrected 2007 data and the 2009 data.

To reduce the inspection task fine scales were not usually considered and many of the data derived by simple calculations was not checked. For the 2007 data only scales larger than the typical residential house were checked. The 2009 data was checked at much coarser scales because the time constraints were more restrictive and the 2007 data was available for comparisons.

For each year a queue of available tiles, and their checked status was maintained. In 2007 the GEM and vegetation mask was checked for all of the tiles. Large mistakes in the DSM were not anticipated so the nDSM and the DSM were checked in only some of the tiles. In 2009 the GEM, vegetation mask, nDSM and DSM were all checked for all of the tiles.

Any errors found were recorded in vector files with each error annotated by a specific keyword. The tables E.1 and E.2 contain the keywords used and suggestions for finding each type of error.

If any errors required correction a new copy of the map was created and assigned an *edited* label and any corrections occurred on this new data set. Any errors that were fixed were moved to a different vector file and also noted in the history document.

It was especially important to track corrections within the overlaps of tiles to maintain cohesive data sets spanning the whole area. This required careful communication between the manual editors by marking overlaps in the queue as either *currently editing*, *edited* or *overlap difference resolved*.

The following sections detail the methods used to fix commission errors in the GEM, correct the vegetation mask in cloud shadows and repair the vegetation mask in large forests.

Erroneous Roofs, Trees and Other Commission Errors in the Ground Elevation

Erroneous roofs, trees and other commission errors were removed by manually digitizing the error in the ground mask and removing it. The re-interpolation of the area was made to match the rest of the ground surface by using a candidate ground set that contained a band of the original GEM surface around the edge. This avoided most boundary issues, however it was noticed that cliffs and true discontinuities within this band received further smoothing.

In more detail, vectors with the following names were drawn around the error:

- *del* Around where the candidate ground image should contain no data. This was used to remove errors in the ground mask.

- *use_gmk* Was drawn around areas where the original ground mask should be used in the candidate ground image.

- *use_cleaned_gmk* Only when both dates of ground elevations were available. Inside this region points in the original ground mask were compared to the GEM from another year and removed if the difference was above a user-specified threshold.

Outside these regions the heights of the current ground elevation was used in the candidate ground image.

Batch scripts and a rasterising program were used to process the above vectors and generate a new ground elevation in a small neighbourhood of the vectors (see appendix E.3, `GEMfix.bat`). This ground elevation patch was then substituted into the full ground elevation data, and a note added to the history document.

Correction of Cloud Shadow in the Vegetation Mask

When time permitted the vegetation mask in cloud shadows was manually fixed. A threshold on a vegetation index found by a directed two group contrast using CVAR proved to work well in these shadows when often no threshold was possible to make NDVI work. The index was

$$\frac{137.627i_1 - 222.611i_2 + 239.899i_3 - 19.320i_4}{-10000 - 3.813i_1 + 23.956i_2 + -37.325i_3 - 7.251i_4}$$

(where i_1=red band, i_2 = green band, i_3 = blue band and i_4 = near-infrared band.)

Once a good threshold was chosen, a boundary vector that minimised unsightly boundary issues was drawn and used to substitute the better vegetation mask into the full data.

Vegetation Mask Problems in Large Forests

In the 2007 data the calibration appeared to be skewed in large forests, which occurred in three locations, Myalup State Forest in the South, Gnangara-Moore River State Forest in the North, and the Darling Range forests in the South East. The calibration differences caused the vegetation mask to include virtually all the bare ground (access roads etc.).

The correction for each area was different, the search for parameters is described in section 3.2, and only the methods used to merge the new masks with the original vegetation mask are described here.

For the Myalup State Forest vectors were drawn around the boundary following access roads, and the origin and tolerance was altered inside to $O_{nir} = 300$, $O_r = -300$, and $tol_{nir} = tol_r = 200$.

In the Gnangara-Moore forest it was assumed that any ground was not green growing vegetation (it was a pine plantation and a blanket of pine needles on the ground prevented other vegetation from growing) so vectors were drawn around the forest and any vegetation shorter than 500mm was removed.

The Darling Ranges forest was much harder to fix, it was a natural forest instead of a pine plantation, so had no well defined boundaries and contained vegetation at ground level. Inside the forest the best origin and tolerance parameters found for NDVI were $O_{nir} = 300$, $O_r = -300$, and $tol_{nir} = tol_r = 200$. The

boundary of the forest was drawn as a piecewise linear curve and the parameters linearly feathered over $4km$ until they matched the parameters for the full Urban Monitor region. Due to the size of the boundary and the resolution of the data the feather distance of 4km was calculated directly from the formula for this boundary.

4.2 Results and Discussion

In both years the largest, most obvious, and most time consuming repairs were commission errors in the GEM caused by forests and industrial buildings. The 2009 corrections benefited significantly from comparison to the corrected 2007 GEM, yet applying the fixes to each individual GEM error was still time consuming.

4.2.1 Summary Vegetation Statistics

Summary vegetation statistics may be quickly generated from this data, on both small and large regions of interest. As an example the total grass area, total tree area, and the mean and variance of tree heights for the full Urban Monitor region for both years are given in table 4.2.

The commission and omission error estimates from section 3.2.2 (tables 3.10, 3.11) were used to calculate upper and lower bounds for these areas.

The tree areas were surprisingly consistent given the large error bounds, suggesting that the phenomena that cause errors (mostly shadows or DSM errors) were equally common in both years.

The largest difference between the years was a loss in vegetation below $0.5m$. This may be due to inaccuracies in the method such as sensor differences, processing differences, or classifier differences, or reflect a real change in the vegetation. If the change was real, it might have been a temporary browning of grass (due to a weather event) or a more permanent loss. More dates of data are required to determine whether this loss was significant.

4.2.2 Evaluation of Distributed Inspection Methods

The nDSM proved a valuable tool for evaluating both the GEM and the DSM. The DSM was difficult to check directly because the variation from valleys and hills was usually much larger than the errors. However in the nDSM cloud induced matching errors and other DSM errors were usually obvious spikes or depressions. Errors in the GEM such as large terrain omissions and mislabelled forest canopy were also obvious as depressions or unrealistically tall protuberances.

Whilst repairing data sets was extremely time consuming, the time required to copy the corrections on overlaps across to neighbouring tiles was also significant. This copying was a relatively straightforward task that could be automated in the future.

In practise a significant portion of inspection time was also spent waiting for images to load; often starting a new tile would take more than five minutes and each new view would take another 30 seconds. It was found that most investigations used fairly consistent viewing parameters across the whole Urban Monitor area. This could be exploited by creating compressed quick-viewing images in jpeg2000 format. The jpeg2000 images would drastically reduce this reading time and could be generated automatically using batch scripts.

Initially none of the maps generated on iVEC had associated history files, this meant that when edited versions of spectral and DSM data was created new history files were created that often did *not* include references to the raw data sets. The histories of these iVEC generated files were later created manually. If time permitted they should have contained information about the programs, parameters, and input data sets.

4.2.3 GEM

In 2007 more than 100 forest errors were fixed and more than 100 more recorded but not fixed. In 2009 the numbers were similar.

Twenty industrial roofs and suburbs required fixing in both 2007 and 2009. These industrial regions were very time consuming to fix due to the number of roofs in each area.

The effects of the piecewise generation of the GEM were minimal with the boundaries noticed only three times.

There were two locations (identical in both years) where large terrain omissions were caused by steep slopes. Repairs of these errors did not occur due to time restrictions.

Large terrain omissions were occasionally caused by plantation forests. In these cases there was no ground elevation information in the DSM. In the future ground elevation data could be obtained from other dates when the forest is sparser or from different sensors (such as LiDAR or historical maps).

Due to time restrictions, the data was not rigorously checked for errors that were the size of residential houses or smaller. In section 2.5.3 twenty-three commission errors were found in a fairly simple $25km^2$ area, suggesting at least 400 unrecorded small-scale errors over the whole region. Some of these errors could be fixed by comparison of GEMs from other years however roof commission errors are likely to persist through multiple dates because most roofs were captured consistently irrespective of viewing direction differences. This was confirmed by the fact that most large roof errors in 2007 were also present in 2009. This meant that most of the errors were caused by mistakes in the ground filter and not by the sensor.

To remove both persistent and ephemeral errors it would be ideal to have a method for automatically merging GEMs between years. One year of repaired GEM data could then be used to fix the GEM errors in every year.

4.2.4 Vegetation Mask

As with the GEM, the vegetation mask worked well in general with the errors found covering a tiny fraction of the entire $9600km^2$ area. The most significant errors found in the 2007 mask (except for the already mentioned forests) were occasional large regions of water labelled vegetation and errors caused by cloud shadows (approximately 100 cloud shadow issues were found in 2007).

A good water mask derived from a combination of spectral, DSM and GIS data could solve the former problem. However the author has no suggestions for the automated removal of cloud shadows. The occurrence of cloud shadows could be reduced by careful selection of data collection times. For example the number of cloud shadows affecting the vegetation mask dropped down to just 9 in 2009.

The 2009 vegetation mask appeared more consistent than the 2007 mask with

only one large water issue and no repairs required in large forests. However the large-scale-only checking of the 2009 data prevents any conclusion to be drawn about small roof commission and tree omission errors.

Blue roofs were consistently labelled vegetation in both years. Whilst designing the vegetation classifier (section 3.2) it was deemed that the low frequency of blue roofs did not warrant a special blue roof rule. However the thorough search of the 2007 mask revealed that in some suburbs blue roofs have a very high frequency. In the future a vegetation classifier that removes blue roofs should be used.

In 2007 50 non-blue roofs were noticed in the vegetation mask. In reality the number was probably much higher because house-sized vegetation errors were not focused on due to time constraints. The cause for this was unknown, but it is hoped that the better calibration of the 2009 data fixed the issue.

Dataset	Description
DSM	The elevation of each pixel in millimetres above the geoid (geoid ≈ sea level)
GEM	The elevation of the ground in millimetres above the geoid (geoid ≈ sea level)
nDSM	The relative height of pixels above the ground in millimetres
Multispectral	Orthorectified multispectral images (near infrared, red, green and blue) calibrated to ground reflectance
Vegetation mask	A classification of the region into green growing vegetation and everything else. Dry heath, grass and brown trees were not labelled as green growing vegetation.
Vegetation height	The height of green growing vegetation above the ground.
Vegetation index	A vigour index, from NDVI, of each vegetation pixel.
Green grass mask	A mask containing all the green vegetation that was below 500mm.
Bushes/trees mask	A mask containing all the vegetation that was above 500mm.
No data mask	A data set marking locations without elevation data, spectral data, or both.
Roof mask	A very conservative labelling of roof pixels for use in the vegetation classification (see section 3.2)
Ground mask	The set of candidate ground points from inflows

Table 4.1: The various maps used and generated for the entire Urban Monitor region.

	2007	**2009**
area captured	$9396km^2$	$9303km^2$
area of vegetation - upper bound	$3670km^2$	$3222km^2$
area of vegetation	$3219km^2$	$2583km^2$
area of vegetation - lower bound	$3106km^2$	$2329km^2$
mean height of vegetation above $2m$	$8.23m$	$8.25m$
standard deviation of vegetation above $2m$	$5.12m$	$5.07m$
area of vegetation above $2m$ *- upper bound*	$1748km^2$	$1890km^2$
area of vegetation above $2m$	$1297km^2$	$1251km^2$
area of vegetation above $2m$ *- lower bound*	$1284km^2$	$997km^2$
area of vegetation between $0.5 - 2m$	$464km^2$	$394km^2$
area of vegetation below $0.5m$	$1494km^2$	$937km^2$

Table 4.2: Vegetation statistics for the entire Urban Monitor region.

Chapter 5

Conclusion and Future Work

In this thesis methods for creating bare earth surfaces and land cover classifications were developed and applied to a $9600km^2$ mosaic of decimetre resolution aerial photography. Apart from the value of this data to the Perth community these methods demonstrate the worthiness of high resolution aerial photography for automated broad region remote sensing. Statistics for summarising fine scale city-wide spatial processes are now possible. The greater spatial information and better monitoring of these spatial processes will be useful in many urban and non-urban fields such as town planning and environmental management, and support further scientific endeavours such as habitat analyses and model validation.

Two different techniques were investigated for the land cover classifications (chapter 3), terrain illumination normalised indices trained using CVAR were explored for separating multiple spectral classes whilst for simpler vegetation/non-vegetation classifications a threshold on NDVI was used. The classes derived from these spectral, pixel-based classifiers were then further separated according to the height above the ground (the nDSM). The vegetation class could thus be divided into vegetation below 0.5m (grass), vegetation above 2m (trees) and vegetation above 20m (tall trees) or any height desired by a user.

The GEM was estimated from the DSM using a segmentation/morphological filter and repeated surface fitting with removal of rough data points (chapter 2). This new algorithm produced good estimates in well-vegetated dense suburbs, new suburbs, peri-urban areas and farmland, however it failed on a few rarer terrains: industrial buildings, dense forests, extremely steep hills, sharp depressions and discontinuities. The GEM is important for flood modelling, hydrological

analysis and many other domains. It is required to calculate the nDSM which is an essential tool for many urban land cover classification algorithms, and valuable for error detection, visualisation and the generation of fine grain height statistics.

The generation of the GEM and vegetation classification over the full $9600km^2$ region is described in chapter 4 and was non-trivial due to the volume of data.

5.1 Land Cover Classification

Two different types of classifiers were investigated, one for only detecting green-photosynthetically-active vegetation and the other for detecting all of the common urban cover types. Classes were first extracted from multispectral data using NDVI or CVAR-derived indices. These were then separated by height using the nDSM. In the case of the NDVI vegetation class a morphological rule using the nDSM was also applied.

CVAR indices were trained and tested in three different regions with the purely spectral multiclass classifications achieving $77.5\% - 81.6\%$ accuracy depending on the number of classes and complicating factors such as fire scars and cloud shadows. In all three regions general classes of brown cover, grey cover, photosynthetically-active vegetation and shadow were produced. Sometimes further classes such as water, blue objects and synthetic green cover were also separated. Once separations from the nDSM to extract roofs and trees were introduced the accuracies decreased to between 74% and 80.8%.

CVAR required careful use of the training data and was sensitive to some search parameters. For easy use in the future section 3.1.6 presented a summary of strategies for achieving high quality separations and suggestions for analysing the results. Due to time constraints some of these suggestions have not been tested, but it is expected that they will lead to classifiers that require fewer stages and function over much larger regions.

Due to the sensitivity of the current CVAR methods to variation in spectral values a multiclass classification covering the full Urban Monitor region was not generated; there was not enough available time to choose training sites, apply CVAR and create a classifier for the many strata required.

In comparison the NDVI-based vegetation-only classifier produced good results over the whole region for both years with very few strata. This classifier

used a threshold on NDVI and a morphological rule to remove some roof errors. In the areas with manually interpreted points the accuracy was greater than 90% for both 2007 and 2009, and a thorough visual inspection (in chapter 4) showed similar performance in the remaining areas.

This project has revealed a promising avenue of research into the design of CVAR. Experiments and theory suggested that CVAR with a denominator restricted to be positive would produce better, more reliable separations. A natural implementation of this would be to keep the coordinates of c_2 positive and set $a = 0$, which also has the benefit of following topographic illumination normalisation theory (as described in [47]) more faithfully. This option also reduces the dimension of the search space; due to the scale invariant nature of CVA, the magnitude of c_2 can be restricted to $\|c_2\| = 1$.

In the future the abilities of CVAR will be investigated for classifying consistently over multiple dates. One of its biggest benefits is that it generates features that are good enough that a few features combined with simple operations are sufficient to form quality classifications, which will make it easier to create consistent classifiers over multiple dates. There is also much scope for fusing CVAR features with morphological or textural features to improve the quality of the thematic maps generated in this thesis.

5.2 Ground Elevation Model

The new GEM algorithm developed in this thesis was a hybrid of a segmentation/morphological filter and a surface fitting filter. It required stratification of the region into just two strata, one for the flat coastal plains of Perth and one for the hillier Darling Ranges.

The quality of the GEM was assessed via a ground/non-ground classification using a $300mm$ threshold on the nDSM. Compared to a manually interpreted array of points the classification achieved an accuracy of 89.6%. In reality the accuracy was probably higher because many of the errors in this comparison were due to DSM inaccuracies

In a simple peri-urban region our GEM contained 40 errors (23 commission, 17 omission) which was 40% fewer errors than Inpho's Match-T algorithm. In more complicated or urbanised regions it is likely that the difference will become

even more significant. However the presence of 40 errors in our GEM for this simple area confirmed that improvements are still possible.

Ideally many of these would be fixed without the use of multiple dates of data. Our filter can be thought of as a combination of three operations, a segmentation/morphological filter, a surface fit operation and a roughness filter (with the last two operations repeated many times); essentially we have used surface fitting to communicate information about candidate ground points across space so that outliers may be removed with a local operation (the roughness filter). Thus improvements in the GEM could easily be gained by using different filters that make greater assumptions on the shape of non-ground objects, such as the top-hat filter [67] or another segmentation/clustering/morphological filter.

However the smoothing of cliffs, terraces and other discontinuities can not be fixed by using different filters, and will require the surface fitting to account for these discontinuities. Our thin-plate-spline surface fitter is already capable of fitting in the presence of break points, however an automatic method for the detection of discontinuities is still required. This task was obstructed by spurious edges from objects, occlusions (e.g. trees over cliffs) and the large volume of data.

Appendix A

Uniqueness and Existence of the Discrete Surface Fitting Problem

This proof follows the results in [91, Theorem 5.2 and Ch.5] to prove the existence and uniqueness of the discrete finite element surface fitting problem. Theorem 5.2 states conditions for existence and uniqueness of the solution, the only condition that is not straight forward for the discrete case is $V - ellipticity\ of\ a(.,.)$. The $a(.,.)$ comes from the energy functional, in our case it is

$$\begin{aligned} a(v,u) = &\sum_{i,j} \rho(i,j)\left(v_{xx}(i,j)u_{xx}(i,j) + v_{xy}(i,j)u_{xy}(i,j) + v_{yy}(i,j)u_{yy}(i,j)\right) \\ &+ \sum_{i,j} \alpha(i,j)u(i,j)v(i,j). \end{aligned}$$

Note that in the discrete context the 'derivatives' like $v_{xx}(i,j)$ are all given by sums of neighbours (e.g. $v_{xx}(i,j) = v(i+1,j) - 2v(i,j) + v(i-1,j)$). $V-ellipticity$ means there exists a constant $k \neq 0$ such that $a(v,v) \geq k\|v\|^2$.

Since $a(.,.)$ is bilinear this is equivalent to showing the minimum of $a(.,.)$ on the unit circle is greater than 0, which is equivalent to $a(v,v) = 0 \iff v = 0$.

From here it is pretty straight forward - all the terms in $a(v,v)$ are positive, so when $a(v,v) = 0$ either v_{xx}, v_{xy}, and v_{yy} are 0 or $\rho(i,j) = 0$, similarly for $\alpha(i,j)$ and $v(i,j)$. When the double 'derivatives' are zero differences between pixels along rows and columns must be constant except wherever $\rho(i,j) = 0$ where changes can occur.

Three non-collinear pixels with $\alpha(i,j) \neq 0$ force these differences to be 0 up to $\rho(i,j) = 0$ points. An image that is divided into regions by curves of $\rho(i,j) = 0$

will require 3 non-collinear height constraints in each of the regions, and possibly at any corners of the $\rho(i,j) = 0$ lines.

So for an area divided into regions by curves of discontinuities $\rho(i,j) = 0$ a solution to the discrete surface fitting problem exists and is unique if there are three non-collinear depth constraints in each region, and depth constraints at every corner of the discontinuity curves.

For example in our case where $\rho(i,j) = 0$ only on the edge pixels of the image, 3 non-collinear height constraints forces $v = 0$ on the interior and the straight edges, but at the corners of the image v is not necessarily 0 because the neighbouring pixels above/below and left/right have $\rho(i,j) = 0$ too. The corners are forced to be 0 only if their data matching weight $\alpha(i,j)$ is non-zero. So a truly unique surface exists only if $\alpha(i,j)$ is non-zero at the corners too.

In practise three non-collinear restraints exist in all but the most unusual data sets, and the corner pixels can be set to the same values as neighbours to produce a high quality surface.

Appendix B

Experiments for the generation of ground elevation models

B.1 The Number of Levels and the Behaviour of a Gauss-Seidel Relaxation

The candidate ground sets shown in figure B.1 were each fitted using a cascadic algorithm that transferred to the next level whenever

$$\|\Delta u\|_\infty = \max_{(i,j)\in image} |u_{old}(i,j) - u_{new}(i,j)| < 70.$$

Pyramids with between 1 and 11 levels were tested. Table B.1 summarises the results, for the investigation in the behaviour of Gauss-Seidel relaxations the energy of the surface ($\mathcal{E}_s$ and $\mathcal{E}_d$) and snapshots of the energy distributions ($\mathcal{E}_s(i,j)$ and $\mathcal{E}_d(i,j)$) were recorded after each relaxation. For some surface fits the surface after each relaxation was also exported so that the change in the surface could be investigated.

The sparse data sets (figures B.1e,B.1f) generally underwent significantly more relaxations than the denser data sets. The surface fitting on dense data sets typically used less than 10 relaxations on each level making the results very sensitive to changes in the number of levels.

To check that the relaxation amounts generated by the $\|\Delta u\|_\infty$ threshold were appropriate the number of level 5 relaxations were forcibly increased/decreased on a data set containing a mix of dense and sparse control points (figure B.1d).

Despite this the relaxation amounts of the dense data sets were deemed too low to be relevant to the final surface fitting algorithm.

B.1.1 Number of Levels

For dense data sets multiple levels fully populated with depth constraints were a detriment to the fitted surface, although sometimes the algorithm was much faster. In fact for the dense data set, figure B.1c, a pyramid without any fully populated levels produced the best surface fit.

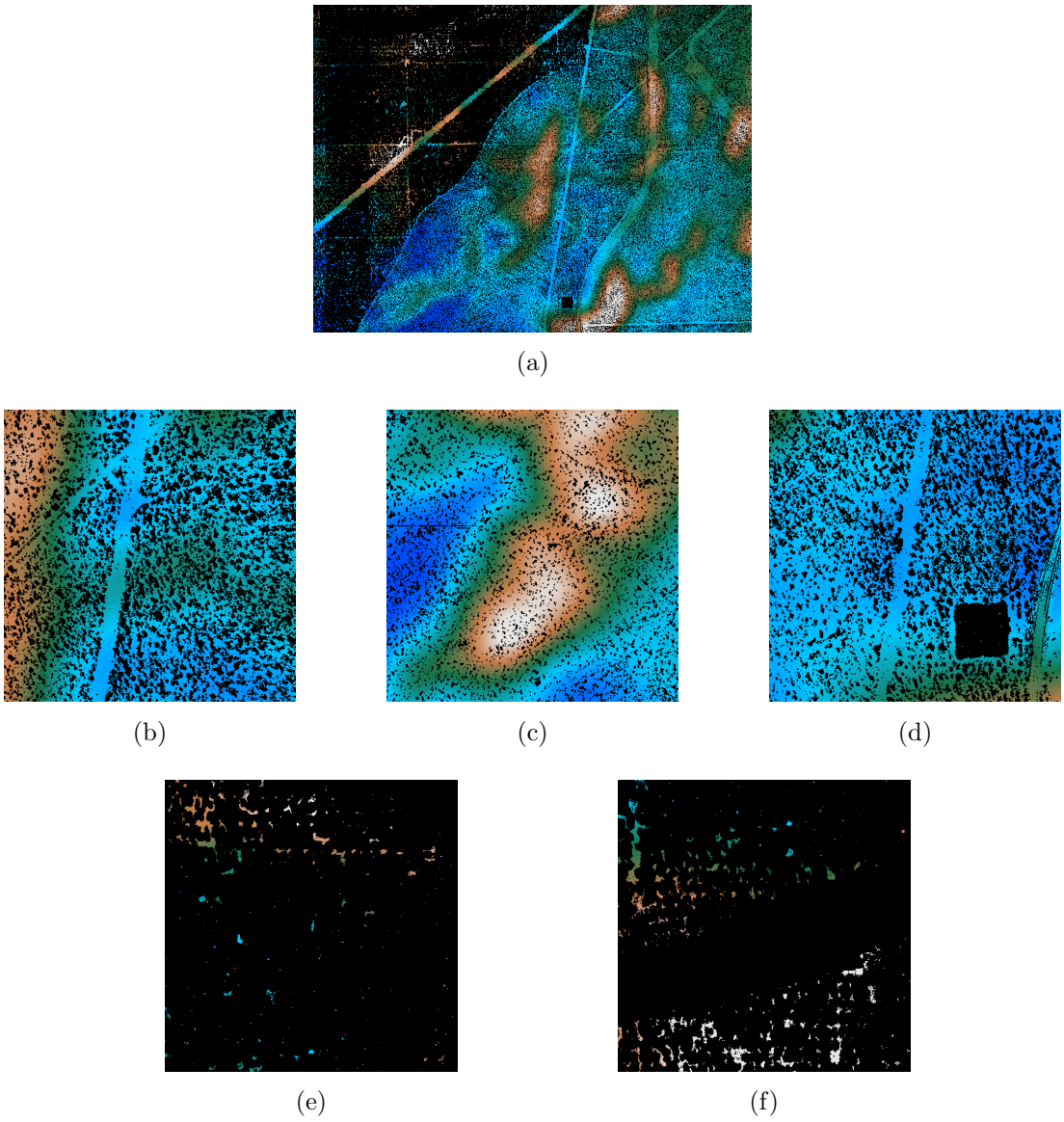
In contrast for one of the sparse data sets (figure B.1f) the surface fits performed better when the coarsest level *was* fully populated with depth constraints.

The abnormally low relaxation of the dense data sets, means only the sparse data set records were relevant so the best number of levels according to this experiment would be the number such that at least one level is fully populated by depth constraints. However more data is required to make a definitive empirical recommendation because this conclusion depends on the results from just one data set. Further testing should involve dense data surface fits with more relaxations and more tests on sparse data.

B.1.2 Effect of a Relaxation

From the logs of the surface fitting, an increase in the total roughness $\mathcal{E}_s$ and a decrease in the data matching penalty $\mathcal{E}_d$ occurred after every relaxation, except for those at the coarsest level. The snapshots of the energy distributions showed that the roughness of the surfaces increased quickly near control points at the start of each level. This confirmed that initial surface guess at each level was too smooth (except for the coarsest level).

The roughness increased near *all* control points, irrespective of their isolation, thus the roughness near patches of inflows commission errors *increased*. Investigation of the actual surface changes showed that the surface became closer to the isolated control points with each relaxation.



(a)

(b)

(c)

(d)

(e)

(f)

Figure B.1: The input candidate ground points used to test the algorithm. All of the smaller data sets are subsets of the first data set (figure B.1a). Hot colours correspond to high elevations, cool colours are low elevations. Black pixels are not candidate ground points, these are points that have been removed from the DSM by inflows or manually. These data sets were chosen to represent a variety of control point densities.

Input figure	**Levels**	**Forced Relax-ations?**	**Relaxations per level (coarsest to finest)**	$\mathcal{E}_s(u) + \mathcal{E}_d(u) = \mathcal{E}(u)$	**Comments**
B.1b	9	None	***11,12,10***,5,5,6,7,10,8	3.305E+8 + 1.538E+9 = 1.869E+9	*The 6 level pyramid was quicker and produced a better energy than the 9 level pyramid.*
B.1b	6	None	***5***,4,6,7,9,8	2.988E+8 + 1.262E+9 = 1.561E+9	
B.1c	9	None	***13,15,12,7***,3,2,3,3,2	1.677E+8 + 3.554E+10 = 3.571E+10	*The 6 level pyramid produced a much better energy but took 0.5 times longer than the 9 level pyramid.*
B.1c	6	None	***2***,2,2,2,3,4	1.926E+8 + 1.417E+10 = 1.437E+10	
B.1c	5	None	24,2,2,4,4	1.416E+8 + 5.216E+9 = 5.358E+9	*The 5 level pyramid achieved a much better energy than both and took very slightly longer than the 6 level pyramid.*
B.1c	1	None	845	6.767E+11 + 1.527E+11 = 8.294E+11	*The single level pyramid took 200 times longer than the other pyramids and produced a much poorer surface that didn't interpolate across the data-less regions.*
B.1e	9	None	14,58,95,79,73,65,60,54,49	2.371E+8 + 1.043E+8 = 3.414E+8	*For this sparse data there was no real difference between a 9 or 10 level pyramid.*
B.1e	10	None	***9***,14,57,95,79,73,65,60,54,49	2.371E+8 + 1.043E+8 = 3.414E+8	
B.1f	9	None	50,20,52,79,106,72,63,54,46	5.108E+8 + 2.943E+8 = 8.051E+8	*For another sparse data set the 10 level pyramid performed better than the 9 level pyramid and no further improvement was gained from the 11 level pyramid.*
B.1f	10	None	***8***,18,20,52,79,106,72,63,54,46	5.109E+8 + 2.644E+8 = 7.753E+8	
B.1f	11	None	***1,15***,18,20,52,79,106,72,63,54,49	5.109E+8 + 2.644E+8 = 7.753E+8	
B.1d	9	None	***9***,13,32,51,31,22,22,17,14	4.391E+8 + 6.413E+8 = 1.080E+9	*A 9 level pyramid with the 5th level artificially decreased by 8 relaxations took longer and produced a poorer fit than the unforced relaxations. Increasing level 5 to include 8 more relaxations produced a much better energy, for only a small increase in work units.*
B.1d	9	Lvl 5 to 14	***9***,13,32,51,31,14,23,18,14	4.477E+8 + 6.493E+8 = 1.097E+9	
B.1d	9	Lvl 5 to 30	***9***,13,32,51,31,30,21,17,14	4.327E+8 + 6.306E+8 = 1.063E+9	

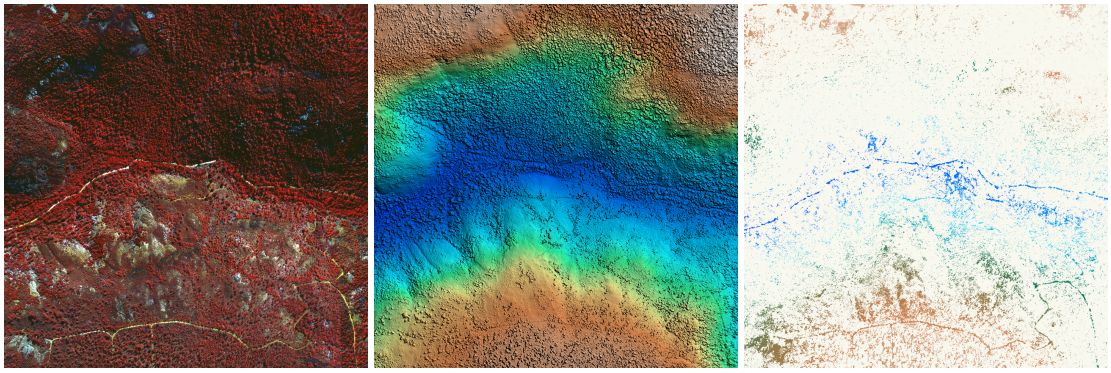
Table B.1: Results of surface fitting with a cascadic algorithm that transferred whenever $\|\Delta u\|_\infty = 70$ was below 70. The bold, italicised levels indicate resolutions at which every node was a control point.

B.2 Outlier Detection Parameter Search

Different roughness thresholds and surface fitting parameters were used for the Darling Ranges and the Perth coastal plain. One site in each region was used to choose the parameters.

B.2.1 Darling Ranges

The test site for the Darling Ranges was a small portion of Karrakup (see figure 1.1 for location). The DSM, a false colour display and the input ground mask can be seen in figure B.2.



(a) false colour display (b) sun-shaded DSM (c) Masked ground

Figure B.2: The region of Karrakup used to guide parameter choices in the Darling Ranges. Hot colours correspond to high elevations, cool colours are low elevations. In the masked ground image the grey pixels are NULL.

Surface fitting and removal of candidate ground points in rough areas was repeated six times in each trial.

In each of the stages the coarsest three levels, corresponding to resolutions of 25m or larger, received many relaxations. This was because relaxations are cheap at these resolutions and it was important to approximate the surface well at large scales.

The initial guess of parameters produced a very poor ground elevation (see figure B.3b and table B.2). The valley was very smooth indicating the removal of too many true ground points, and part of the North West ridge was completely missing. An increase in the relaxations of the first few stages (see trial 2 - table

B.2; no picture shown) did little to change the final surface - in fact the slight increase in roughness from the extra relaxations caused loss of more data in the valley. However further increases in relaxations led to a substantially better surface fit for outlier detection in stage 1 and by the end of all six stages the valley walls were slightly improved (see trial 3 - table B.2; figure B.3c).

Increasing the relaxations of the later stages led to massive loss of the ridge line and no improvement on the valley (trial 4 - figure B.3d). Intermediate data revealed that the first three stages removed most of the data in the valley floor and wall.

Eventually after drastically loosening the energy thresholds of stage 1 (trial 5), stage 2 (trial 6) and stage 3 (trial 7 - see table B.2), and protecting control points below the fitted surface in stage 3 (trial 7 as well) significant amounts of ground in the valley was surviving past stage 3.

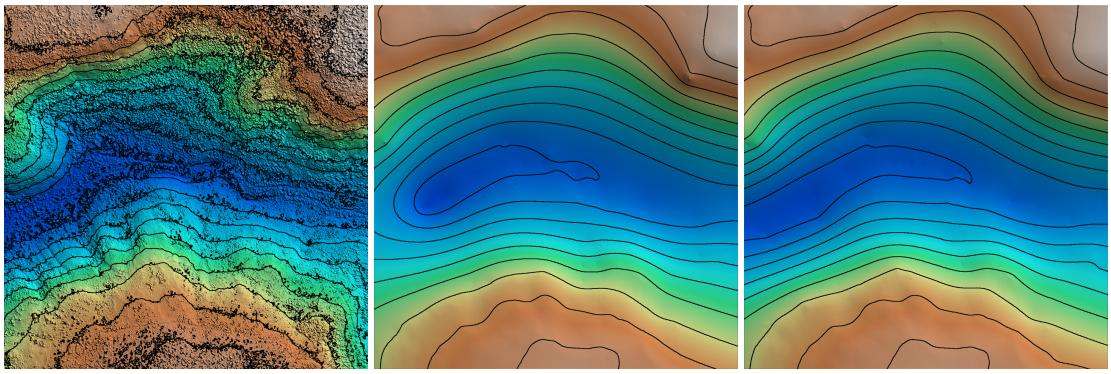
The surface after the six stages of trial 7 was still similar to trial 3 (figure B.3c), but with good amounts of data surviving past stage 3 it was a matter of loosening the energy thresholds for stages 4-6 (trials 8-10 - parameters of trial 10 in table B.2). This trial 10 GEM nicely captured the region, with the biggest error the loss of the creek line in the North West. It is unlikely that this creek line could be improved because the input ground mask does not contain much data near the creek due to the area being extremely steep and/or covered with vegetation.

In the final parameters (see table B.2 and figure B.4) the relaxations were increased significantly to yield a very slight improvement in the quality of the GEM.

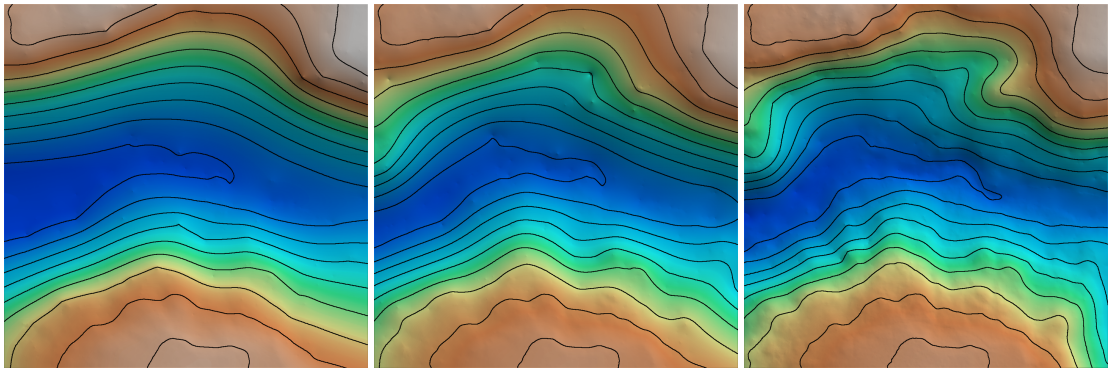
Results The quality of the GEM can be partially assessed by investigation of the difference between the DSM and the GEM (figure B.4b). Common knowledge, such as the height and shape of trees, roads and buildings can be used to assess how well the GEM excluded non-ground objects. The final GEM nicely captured the shape of the southern ridge. The river line was only slightly smoothed over and the North East section of ridge was mostly correct. As mentioned for trial 10 there was a creek line missing (black ellipse in figure B.4b) and there are a few locations where the surface fitting has missed patches of steep bare ground (grey ellipses/polygons in figure B.4b). Neither of these issues will be easy to fix

because they stem from a lack of data in the input ground mask.

The GEM also omitted a sharp knoll (white ellipse in figure B.4b). The partial covering of vegetation and steep slopes meant there was only some data in the ground mask whilst the sharpness of the height changes caused the GEM to completely smoothed over it.



(a) DSM (b) Trial 1 (c) Trial 3



(d) Trial 4 (e) Trial 7 (f) Trial 10

Figure B.3: Karrakup Ground Elevation Versions with a contoured DSM for reference

B.2.2 Perth Coastal Plain

This section describes the parameters tested for outlier removal in the coastal plain. The testing area covered parts of Subiaco and Kings Park (see 1.1). The DSM, a false colour display and the input ground mask can be seen in figure B.5.

The starting guesses for this search were the Darling Ranges parameters (trial 1 - figure B.6a). These parameters yielded a GEM that captured most of the

Trial 1	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	n	y	n	y	y
$\mathcal{E}_s$ thresh.	50	30	10	15	5	5
lvl 9 relax						
lvl 8 relax						6
lvl 7 relax					14	17
lvl 6 relax		14	14	14	27	28
lvl 5 relax	20	27	27	27	40	39
lvl 4 relax	40	40	40	40	53	125
lvl 3 relax	60	170	170	170	183	211
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Trial 2	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	n	y	n	y	y
$\mathcal{E}_s$ thresh.	50	30	10	15	5	5
lvl 9 relax						
lvl 8 relax						6
lvl 7 relax					14	24
lvl 6 relax		**24**	**24**	14	37	42
lvl 5 relax	**30**	**42**	**42**	37	60	60
lvl 4 relax	**60**	**60**	**60**	60	83	140
lvl 3 relax	**90**	**180**	**180**	180	203	220
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Trial 3	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	n	y	n	y	y
$\mathcal{E}_s$ thresh.	50	30	10	15	5	5
lvl 9 relax						
lvl 8 relax						6
lvl 7 relax					14	24
lvl 6 relax		24	24	14	37	42
lvl 5 relax	**40**	47	47	42	60	60
lvl 4 relax	**80**	70	70	70	83	140
lvl 3 relax	**120**	185	185	180	203	220
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Trial 7	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	n	**n**	n	y	y
$\mathcal{E}_s$ thresh.	**250**	**70**	**70**	15	5	5
lvl 9 relax						
lvl 8 relax						**20**
lvl 7 relax					**20**	**36**
lvl 6 relax		24	24	**20**	**45**	**52**
lvl 5 relax	40	47	47	**45**	**70**	**68**
lvl 4 relax	80	70	70	70	**95**	**144**
lvl 3 relax	120	185	185	**185**	**210**	220
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Trial 10	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	n	n	n	y	y
$\mathcal{E}_s$ thresh.	250	70	70	**30**	**30**	**20**
lvl 9 relax						
lvl 8 relax						**30**
lvl 7 relax					**30**	**46**
lvl 6 relax		**38**	**38**	**30**	**55**	**62**
lvl 5 relax	**50**	**59**	**59**	**55**	**80**	**78**
lvl 4 relax	**80**	**80**	**80**	**80**	**105**	**151**
lvl 3 relax	**110**	**190**	**190**	**190**	**215**	**224**
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Final	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	n	n	n	y	y
$\mathcal{E}_s$ thresh.	250	70	70	30	30	20
lvl 9 relax						
lvl 8 relax						**40**
lvl 7 relax					**50**	**60**
lvl 6 relax		**80**	**70**	**60**	**75**	**80**
lvl 5 relax	**90**	**90**	**85**	**80**	**100**	**100**
lvl 4 relax	**100**	**100**	**100**	**100**	**125**	**166**
lvl 3 relax	110	**200**	**200**	**200**	**225**	**232**
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Table B.2: Parameters for Optimisation of Outlier Detection in Hills.
For each trial and each outlier detection phase (labelled *OD*) whether the removal of control points below the surface is allowed, the $\mathcal{E}_s$ thresholds and relaxations used (with level 9 being the finest level (0.2m GSD) and level 0 the coarsest) are listed. Bold values are those that have changed since the previous displayed trial.

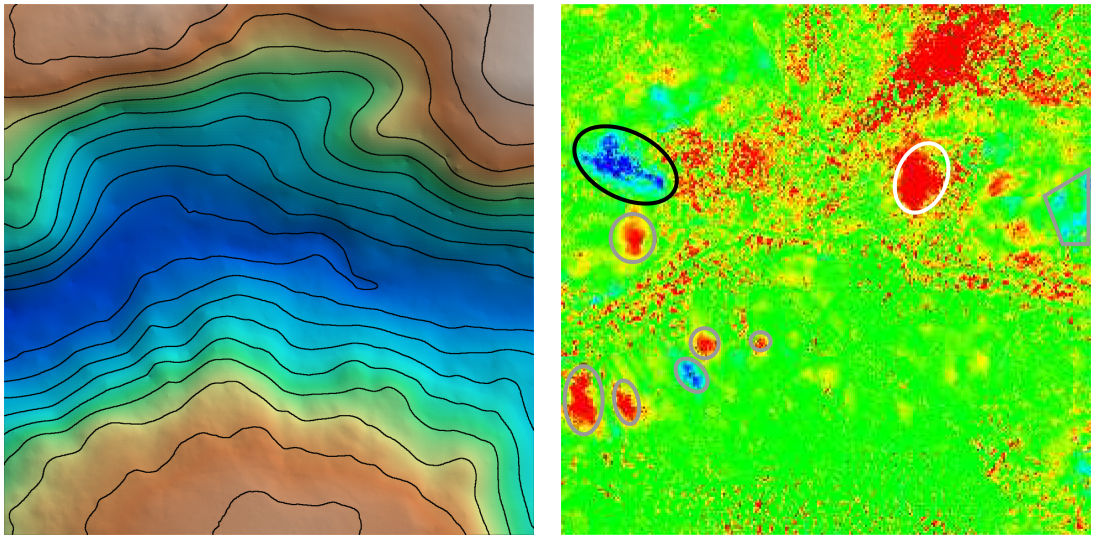


Figure B.4: Final Karrakup Ground Elevation with the difference to the DSM. *Left:* The final GEM with sun-shading and contours (grey = high elevations, blue = low elevations). *Right:* The difference $DSM - GEM$ (hot colours = large positive values, cool colours = large negative values).

landscape but still contained many roof commission errors. Drastically lowering the energy thresholds did not yield much change (trial 2 - figure B.6; table B.3), however switching to coarser resolutions (trial 3) removed many of the medium sized errors. The second stage was allowed to remove below ground errors to reduce the removal of too much good data with the coarser resolutions (preserved low outliers forced nearby good control points to be removed). Unfortunately trial 3 also removed significant amounts of the railway tracks (they were the raised features in the bottom left of trial 1 and trial 2).

Further reductions in the energy thresholds (trial 4) did little to remove the gross errors whilst removed more of the railway tracks. Furthermore when trial 4 was tested on other areas of the coastal plain the results were poorer than trial 3.

The parameters chosen from these experiments were not intended to fix commercial/industrial areas so the final parameters were the trial 3 parameters which worked very well in the suburban and parkland areas, and fairly well in commercial areas.

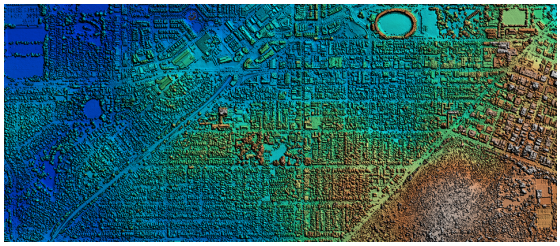
Trial 1	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	n	n	n	y	y
$\mathcal{E}_s$ thresh.	250	70	70	30	30	20
lvl 9 relax						
lvl 8 relax						40
lvl 7 relax					50	60
lvl 6 relax		80	70	60	75	80
lvl 5 relax	90	90	85	80	100	100
lvl 4 relax	100	100	100	100	125	166
lvl 3 relax	110	200	200	200	225	232
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Trial 2	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	n	n	n	y	y
$\mathcal{E}_s$ thresh.	**150**	**40**	**30**	30	30	20
lvl 9 relax						
lvl 8 relax						40
lvl 7 relax					50	60
lvl 6 relax		80	70	60	75	80
lvl 5 relax	90	90	85	80	100	100
lvl 4 relax	100	100	100	100	125	166
lvl 3 relax	110	200	200	200	225	232
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

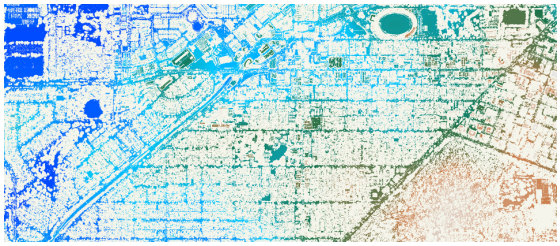
Trial 3	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	**y**	n	n	y	y
$\mathcal{E}_s$ thresh.	**100**	40	30	30	30	20
lvl 9 relax						
lvl 8 relax						40
lvl 7 relax					50	60
lvl 6 relax				60	75	80
lvl 5 relax	90	**90**	**70**	80	100	100
lvl 4 relax	100	100	100	100	125	166
lvl 3 relax	110	110	130	200	225	232
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Trial 4	OD1	OD2	OD3	OD4	OD5	OD6
rmv below?	n	y	n	n	y	y
$\mathcal{E}_s$ thresh.	**50**	40	30	30	30	20
lvl 9 relax						
lvl 8 relax						40
lvl 7 relax					50	60
lvl 6 relax				60	75	80
lvl 5 relax	90	90	70	80	100	100
lvl 4 relax	100	100	100	100	125	166
lvl 3 relax	110	110	130	200	225	232
lvl 2 relax	300	300	300	300	300	300
lvl 1 relax	400	400	400	400	400	400
lvl 0 relax	200	200	200	200	200	200

Table B.3: Parameters for Optimisation of Outlier Detection in Plains. For each trial and each outlier detection phase (labelled *OD*) whether the removal of control points below the surface is allowed, the $\mathcal{E}_s$ thresholds and relaxations used (with level 9 being the finest level (0.2m GSD) and level 0 the coarsest) are listed. The parameters that were changed between each trial are bold. Trial 3 was the parameter set used for the final version.



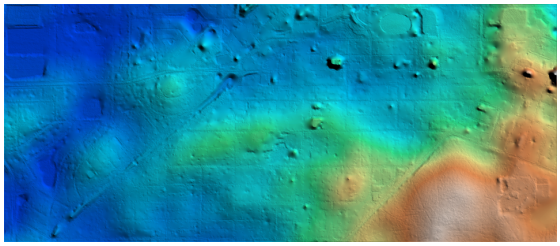
(a) sun-shaded DSM



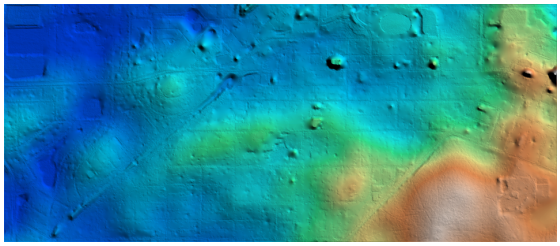
(b) input ground mask

(c) false colour display

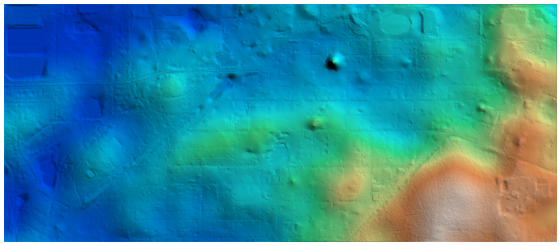
Figure B.5: The Subiaco-Kings Park test region used to guide parameter choices for the Coastal Plains. In the DSM and ground mask grey shades correspond to high elevations and cool colours are low elevations. The NULL values are in grey in the ground mask image.



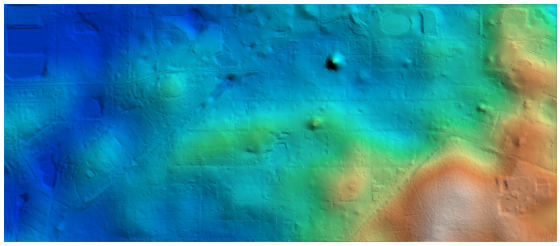
(a) Trial 1



(b) Trial 2



(c) Trial 3



(d) Trial 4

Figure B.6: Subiaco-Kings Park Ground Elevation Versions. Trial 3 was chosen to be the final version.

Appendix C

Detailed description of the GEM generation implementation

The usual work flow to develop a GEM from the DSM was as follows. The inflows segmentation/morphological filter was applied first and the resulting segments combined with the heights in the DSM to create an array of pixels labelled by their elevation or NULL where the pixels were deemed non-ground. From this set of candidate ground points the GEM was generated using the program called `tpsToGriddedPoints` (originally it was called `Terzopoulos`).

The implementation of inflows and Terzopoulos along with the parameters used are described below.

C.1 Implementation of inflows

Inflows was already implemented by Peter Caccetta and is described in [12]. The program inputs were a DSM, an output file name, and four parameters: a scale factor used to convert the values of the pixel heights into the same units as the pixel widths; a slope threshold in degrees that was used to divide the DSM into segments; an area threshold in pixels; and an elevation threshold used to ignore low pixels.

The program first scanned through the image marking pixels as either +, =, or − according to the direction and magnitude of the biggest difference with neighbouring pixels. The regions of pixels labelled = formed the segments that were later chosen to be ground or non-ground.

The image was then rescanned counting the boundary pixels and the direction of the slope at the boundary. Each pixel in a segment was labelled by the percentage of inward flowing pixels to the total number of boundary pixels (pixels below the elevation threshold and edges of the image did not contribute to the total boundary). Those segments below the area threshold then had their value increase by 100.

The final output was between 1 and 100 for pixels in segments larger than the area threshold. For segments smaller than the area threshold the percentage of inward flowing pixels was shifted by 100 units such that the output was between 101 and 200. Pixels below the height threshold, or above the slope threshold were labelled 0.

For the ground filtering the slope threshold was 25^o, the area threshold was 10 pixels, and the height threshold was $0m$. The segments labelled between 50 and 100 were the candidate ground locations and used to mask the DSM. This masked DSM then formed the input to `tpsToGriddedPoints`.

To generate the roof mask for the vegetation classification (section 3.2) inflows was applied to the nDSM with the same slope threshold of 25^o, an area threshold of 1000 pixels and a height threshold of $1m$. The segments between 1 and 50 were kept as roofs.

C.2 Implementation of Terzopoulos

This program was written in C++. The `tpsToGriddedPoints` program required an input ERMapper-Signed32BitInteger data set, a parameter file and an output file name. The different parameter files selected between performing hills or plains outlier removal before performing the final surface fit.

Each parameter file contained a one or more lines, each line corresponding to an operation. The operations available were either

- `cascadicFit` which called `cascadicMultiresFit()` to fit a surface, or
- `SurfFit_roughRemoval` which estimated relaxation amounts from the parameters, called `cascadicMultiresFit()` and then removed rough control points using the function `removeRoughControlPoints()`.

The program recorded the relaxation amounts, the number of work units, the largest change in the surface after each relaxation, any outputs and the elapsed time in a log file.

Some other command line parameters available were:

- A parameter `depthWeightinRelaxation` for altering the data matching weight $\alpha(i,j) = \frac{depthWeightinRelaxation}{A}$. By default $depthWeightinRelaxation = 1$. Values greater than 1 increased the importance of data matching, values lower than 1 increased the importance of smoothness.

- An optional intermediate export directory `-miscExportTo`. Used for some debugging, the program then saved many snapshots of the surface and energy distribution to this directory at various times during the surface fitting. It usually produced a large number of files.

- If a discontinuity map was provided (through option `-discons`) then slightly different versions of `cascadicMultiresFit()` and `removeRoughControlPoints()` would be called to cater for the presence of discontinuities in the interior of the image.

The version used for widespread application in chapter 4 was called `Terzopoulos` (version `1.0.5.34`).

Although the input and outputs were in integers the program internally stored the data as floating points. This prevented the relaxations from prematurely converging due to truncation errors.

The edge pixels of the image were set to NULL on import because originally outliers on the boundary could not be removed using roughness ($\mathcal{E}_s(i,j)$ was always 0 on the boundary because the original discontinuity parameter ρ was set to 0 on the boundary) This removal of edge control points still occurred in the final version even though the new discontinuity parameters $\rho_x, \rho_{xy}, \rho_y$ meant that $\mathcal{E}_s(i,j)$ was only 0 at the corners of the image.

C.2.1 cascadicMultiresFit

The `cascadicMultiresFit()` function accepted as inputs the set of candidate ground points, the number of levels to use, the relaxations at each level and some

parameters for exporting intermediate data. It returned a surface fitted using a cascadic multiresolution approach.

The function first generated the pyramid of control points iteratively where coarse data pixels were created from the fine level by averaging nearby control heights. The grids of each level were offset such that coarse nodes were at the centres of the squares created by the next finer level nodes. A coarse node was given a NULL control height if none of its four fine-level neighbours had a control height. Otherwise the control height was the average height of the four neighbours.

The coarsest level surface was then initialised to the coarse heights and any node that wasn't a control point was initialised to the average height of the coarse data.

The allotted number of Gauss-Seidel relaxations were then performed and the surface transferred to the next finer level. The the direction of iteration was alternated for each relaxation to replicate symmetric Gauss-Seidel relaxations.

The relaxation iterated through the image setting the height $u(i,j)$ of each pixel such that

$$
\begin{aligned}
0 &= \frac{\partial(\mathcal{E}_s(u) + \mathcal{E}_d(u))}{\partial u(i,j)} \\
&= \big(u(i,j) - 2u(i-1,j) + u(i-2,j)\big)\rho_x(i-1,j) \\
&\quad + \big(-2u(i+1) + 4u(i,j) - 2u(i-1,j)\big)\rho_x(i,j) \\
&\quad + \big(u(i+2,j) - 2u(i+1,j) + u(i,j)\big)\rho_x(i+1,j) \\
&\quad + \big(2u(i,j) - 2u(i-1,j) - 2u(i,j-1) + 2u(i-1,j-1)\big)\rho_{xy}(i-1,j-1) \\
&\quad + \big(-2u(i+1,j) + 2u(i,j) + 2u(i+1,j-1) - 2u(i,j-1)\big)\rho_{xy}(i,j-1) \\
&\quad + \big(-2u(i,j+1) + 2u(i-1,j+1) + 2u(i,j) - 2u(i-1,j)\big)\rho_{xy}(i-1,j) \\
&\quad + \big(2u(i+1,j+1) - 2u(i,j+1) - 2u(i+1,j) + 2u(i,j)\big)\rho_{xy}(i,j) \\
&\quad + \big(u(i,j) - 2u(i,j-1) + u(i,j-2)\big)\rho_y(i,j-1) \\
&\quad + \big(-2u(i,j+1) + 4u(i,j) - 2u(i,j-1)\big)\rho_y(i,j) \\
&\quad + \big(u(i,j+2) - 2u(i,j+1) + u(i,j)\big)\rho_y(i,j+1) \\
&\quad + \big(u(i,j) - d(i,j)\big)\alpha(i,j),
\end{aligned}
$$

which is a constant complexity calculation that can be thought of as a kernel multiplication involving the heights of nearby nodes.

The boundary of the image was modelled by assigning the edge pixels discon-

tinuities that were hard coded into the relaxation calculations. On the vertical edges $\rho_x = 0$, on the horizontal edges $\rho_y = 0$, and $\rho_{xy} = 0$ on all of the boundary.

The finite elements used were the six-point full quadratic interpolants used by Terzopoulos' and described in [90]. The height of the surface within each square $[i, i+1) \times [j, j+1)$ was given by a second order polynomial that matched the grid heights at the four corners *and* at two extra points $(i-1, j)$ and $(i, j-1)$. In [90] it was shown that with increasingly fine grids these elements converge to the unique solution of the continuous problem. On the boundary or near discontinuities this polynomial was restricted, cut in half, or even set to constant according to $\frac{\partial \mathcal{E}_s(u)}{\partial u(i,j)}$. In practise this polynomial was mostly used by the transfer function to calculate the heights of the coarse surface at the location of the fine nodes.

Within the code there were a number of different functions available for manipulating and querying a particular level. This included the required forward and backward Gauss-Seidel relaxations, calculation of both the total and point-wise $\mathcal{E}_s$ and $\mathcal{E}_d$, and exports of intermediate energy maps.

C.2.2 removeRoughControlPointswDiscons

The outlier removal function `removeRoughControlPointswDiscons()` received the set of candidate ground points, a fitted surface, the smoothness threshold and the choice of removal or preservation of outliers below the surface.

The smooth energy

$$\mathcal{E}_{s(i,j)}(u) = \frac{1}{2A}\left(\rho_x(i,j)u_{xx}^2(i,j) + 2\rho_{xy}(i,j)u_{xy}^2(i,j) + \rho_y(i,j)u_{yy}^2(i,j)\right)$$

at every node (i, j) was calculated. For each node above the smoothness threshold and (when relevant) above the fitted surface the corresponding height control data was removed from the set of candidate ground points.

Appendix D

Functions and Programs for Land Cover Classifications

D.1 Implementation of CVAR

This section contains a description of the individual components of the CVAR analysis (section D.1.1) and the CVAR function itself (section D.1.2).

Some of the tools were useful outside the CVAR analysis, for example to apply already-derived CVAR transforms to another set of pixels, or to create new plots.

D.1.1 Components for CVAR

Creating the Training Data

A program `ExtractPixelsSupper.exe` has been written to extract the spectral values of pixels from ERMapper raster files (data stored in band-interleaved-by-line format with the header information in a different file with extension '.ers'). It was designed such that the training sites are listed as boxes in the ERMapper vector. Thus training sites can be drawn using ERMapper's vector drawing capacities. Each site must have a unique name to make the site-wise separations properties, and should have a standard format (e.g. `dt_br:56`) to make directed separations easier to apply.

The list of box objects (note that ERMapper saves the vectors as human-readable lists) is scanned by `ExtractPixelsSupper.exe`. The program extracts the spectral values of every pixel that lies completely within each box and writes

them to a text file, along with the box's name.

The program can read values in unsigned 8 bit integers, signed 16 bit integers and signed 32 bit integers. The output was a text file where each line was a list of the values of a pixel followed by the name of the box that the pixel was in.

The program can also extract the value of pixels corresponding to *point* objects which was useful for extracting the labels from the automated classifiers and generating the confusion matrices.

The list of pixel values and site names was imported by the CVAR functions.

Import the training data

The import function `loadpixelInfoToFrame()` read the list of training pixel values created by `ExtractPixelsSupper.exe`, and in conjunction with a *user specified function* `sitename2groupname()` created a data frame of the training pixels, with a factors corresponding to the site name, and another factor for the group name.

The `sitename2groupname()` function must be defined previously by the user. The input is an index for the site, which is automatically generated by R when it realises that the site name column is of factor type. It must return the group name.

A site-wise CVAR analysis is obtained by a 1-to-1 mapping from the site name to the group name. Multiple sites were assigned the same group by using some list or conventions within the site names. The special group name `NOT_IN_USE` instructs `loadpixelInfoToFrame()` to omit those training sites from the data frame (which can be useful for removing extreme sites or groups).

Translation of the origin and removal of particularly dark pixels

The function for shifting the origin and removing particularly dark pixels `cleanOutNegativesANDShiftOrigins()` accepts a list of tolerances and origins for the bands. All pixel values were then translated to make the origin 0 with any negative values truncated to 0. Any pixels such that all bands were below their respective tolerances were removed.

CVA functions

The functions for calculating the between-group **B** and within-group **W** sum-of-squares matrices were `betweenMatFramePlStdMD()` and `withinMatFramePlStdMD()` respectively.

The `withinMatFramePlStdMD()` used the standard covariance function

$$\mathrm{Cov}(X) := \mathrm{E}[(X - \mathrm{E}[X])(X - \mathrm{E}[X])]$$

(where everything is scaled by by $n - 1$ instead of n[73]).

The function `cvaRootCampbellv2()` used these two matrices to calculate the solution to CVA problem. First both **B** and **W** were scaled by a diagonal matrix E chosen such that the diagonal elements of the new within matrix $\mathbf{W} = E^T \mathbf{W}_{old} E$ have a magnitude of one.

The canonical vectors for CVA were calculated by solving the eigen-problem given in [19]

$$\text{find } v \in \mathbb{R}^4, \lambda \in \mathbb{R} \text{ such that } (D^{-1/2} U^T B U D^{-1/2} - \lambda I)v = 0.$$

Where D and U are the eigenvalues and eigenvectors of $UDU^T = \mathbf{W}$. However the solution vectors v to the above equation are not in the starting $\frac{x}{a+c_2 \cdot x}$ space and needed pre-multiplication by $EUD^{-1/2}$ to obtain the vectors that minimise the between-groups and within-groups ratio.

Following the suggestion of [17] we ignored any eigenvectors of **W** with eigenvalues close to zero. Thus if **W** has a single eigenvalue of zero then U becomes a 4×3 matrix and D becomes a 3×3.

`cvaRootCampbellv2()` returned an object containing a list of eigenvalues, and a matrix of normalised eigenvectors, where the columns corresponds to the eigenvectors.

Objective functions for the simplex search

The objective function used in the simplex search `objFirstEigenVal()` first weighted the data frame according to the input c_2 and a to produce a data set of $\frac{x}{a+c_2 \cdot x}$ values, calculated the sum-of-square matrices and returned the largest final canonical root (eigenvalue) from `cvaRootCampbellv2()`.

For the CVAR16 variant a different objective function that required an input vector long enough to generate a different c_2 for each band used.

The calculation of this objective function was extremely quick because the training data sets were fairly small so the CVA canonical root was extremely quick to calculate.

Simplex search

The Nelder-Mead simplex search [68] was implemented in `simplexSearch()`, it required the training data, a matrix of starting vertices (a set of guesses for c_2), a, the choice of objective function, and the usual simplex search parameters.

For all the results presented, the reflection coefficient was set to 1, the expansion coefficient set to 2 and the contraction coefficient set to 0.5, whilst the tolerance was usually it was set to 1×10^{-6} or smaller.

The simplex search exited when the objective of all the vertices were within the tolerance of each other. However sometimes the search would reach a point where the simplex stopped evolving before reaching tolerance, in this case the best current vertex is returned with a warning.

Plotting Tools

Functions for plotting pixels, the site names and group names, coloured by group in either 2D or 3D are included. The site names and group names are placed at the mean of the site/group respectively.

At this point there functions that allow the cursor to draw discrimination boundaries or feature vectors which then automatically create strings suitable for pasting into other ERMapper and other strings.

D.1.2 The CVAR function

The CVAR function `CVAR4` gathers the above tools together to import the training data from the output of `ExtractPixelsSupper.exe`, and execute the CVAR analysis through to plotting the separation of the first two CVAR transforms.

It accepted a choice of a, the tolerance for simplex search, the size of the initial simplex (which should be smaller than the bound given in section 3.1.5), the origins and tolerance of the spectral values, and a file for exporting the final c_2 and CVA vectors.

These parameters were all optional, however the defaults were designed to be backwards compatible and do not always represent the best values. The only required inputs were the file name of the training data text file (output from `ExtractPixelsSupper.exe`) and a *hidden* input: the `sitename2groupname()` function must be defined for importing the training data into R.

The `CVAR4` function first imported the training data, shifted the origin and removed pixels below tolerance (when tolerances and origins were supplied). The vertices of the initial simplex were then randomly generated from a uniform distribution between -1 and 1, and scaled by the simplex size parameter. The Nelder-Mead search used this simplex and the objective function `objFirstEigenVal()` to search for the best possible c_2. The reflection coefficient was set to 1, the expansion coefficient set to 2 and the contraction coefficient set to 0.5.

The time required for the simplex search usually varied from 30 seconds to a few minutes depending on the size and distribution of the training data, and the search tolerance.

On completion the c_2 and a were used to weight the training data to produce a data set of $\frac{x}{a+c_2 \cdot x}$ values. The sum-of-squares matrices were then calculated, and the CVA roots and vectors derived. With all components of the CVAR canonical transforms calculated the transforms were applied to the training data. The first and second CVAR canonical transforms were then plotted.

The output from `CVAR4()` was a list object containing the raw training data, the cleaned and shifted data, the final c_2, the weighted training data, the CVA results, and the training data transformed to the CVAR canonical transforms. For pasting into the ERMapper formula field string versions of the CVAR transforms were also generated.

Other auxiliary information was also included in the output, such as the name of the training data file, the starting simplex, the number of iterations of the simplex search and the search tolerance used.

D.2 Accuracy Analysis of Classifications

The program `ExtractPixelsSupper.exe` and function `loadpixelInfoToFrame()` both behave well with point data. Thus the accuracy assessments were made by manually interpreting collections of ERMapper point objects.

The automated labels were extracted from the thematic maps using `ExtractPixelsSupper.exe` which resulted in a list of automated class labels and manual descriptions. These were imported into R using `loadpixelInfoToFrame()` and confusion matrices generated. A function, `accNumsFromContTable()`, was used for producing the accuracy statements from confusion matrices (it can be found in `accuracyAssFuncs.R`).

The speed and comfort of manual interpretations were improved through R functions that iterated though a list points, presenting a collage of different displays for each point. A function first used some ERMapper algorithm templates to generate a batch script that created various viewing algorithms for each point, compressed the views to jpeg2000 (using ERMapper's jpeg2000 command line compression tool) and combined them into a collage using the ImageMagick *convert* and *append* tools (see an example collage in figure D.1). Another function was then iterated through the collages and waited for labels from the manual interpreter.

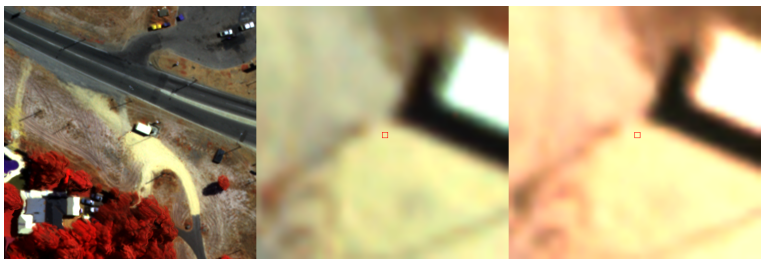


Figure D.1: An example collage created to assist with the manual interpretation of points. *Left:* a false colour display $100m \times 100m$. *Centre:* a false colour display $5m \times 5m$. *Right:* a true colour display $5m \times 5m$. The pixel awaiting manual interpretation is marked with a red point, or red square.

Appendix E

Appendix for Application to Full Urban Monitor Region

Contained in this chapter are descriptions of the different errors recorded (section E.1 and tables E.1 and E.2), detailed instruction for the creation of each of the data sets for the full urban monitor region (section E.2), and a description of each program used (section E.3).

The tiles and file naming conventions may be found in [13, section 4].

E.1 Detection of Errors

Tables E.1 and E.2 describe the different elevation and vegetation errors found. The keywords used for recording them, some techniques that were used to detect them and the typical action taken by manual interpreters are also included. Another type of error, for corrupt spectral data (keyword: *specErr*), is not included in these tables.

Often small house-sized, isolated errors were not recorded due to time constraints. If the error was much larger and more obvious, or if there was a high density of errors then it was recorded. Some errors (with action *repair if massive*) were only repaired if the error was visible when displaying the entire tile.

Other errors (e.g. over-smooth GEMs in sharp landscapes) were not recorded due to the difficulty in detecting them and recording the effected area.

For manual detection of errors in the elevation data a sun-shaded GEM or nDSM, and a false colour multispectral view were often displayed. For the in-

spection of the vegetation maps in urban regions the sun-shaded nDSM was often displayed with the vegetation mask at 70% transparency. For the inspection of the vegetation maps in rural regions a false colour multispectral view was often displayed with the vegetation mask.

The vegetation mask was rarely repaired due to the difficulty in generating a new mask (by searching for new parameters, or using a different index) and combining it with the original vegetation mask. This sort of correction may be possible in the future with a program that slowly changes the parameters with distance from a given boundary vector (a similar method was already used to repair the Darling Ranges - see section 4.1.3).

E.2 Execution of Programs for Data Creation

A table summarising the data sets generated and the format that each used is presented in table E.3.

GEM

1. `winflows`[1] was run with a slope threshold of 25, an area threshold of 10, a scale factor of 1000, and where applicable an elevation threshold below the lowest possible ground point.

2. In the output from `winflows` all segments between 51 and 100 (inclusive) were kept as ground and used to mask the DSM.

3. Run `tpsToGriddedPoints` (originally called `Terzopoulos`) with the Perth coastal plain or Darling Ranges parameters depending on location.[2]

nDSM

1. The program `nDSMfromGEMnDSM` was used. It calculated $DSM - GEM$, and wherever $DSM - GEM < 0$ it set the output to 0.

[1] this is a special implementation of *inflows* from section 2.2 that functions on large datasets

[2] The creation and feathering of the 20000×20000 pieces is not described here. For the full region these steps were carried out by Drew Devereux.

Issue	***Keyword*** **/Action**	**Suggestions for Manual Detection**	**Repair**
An isolated roof in the GEM	*rfInGEM* repair	View the GEM with sun shading from a sun at 45^o elevation, zoomed out so that extents are greater than $2km \times 2km$	See section 4.1.3
Industrial area with many roofs in the GEM	*indRfsGEM* repair	Same as *rfInGEM*	See section 4.1.3
Forest with a poor GEM	*forestGEM* repair	Same as *rfInGEM*, but look for small, sharp peaks in areas that are otherwise very smooth.	See section 4.1.3
GEM missed sharp peaks in dunes, man-made landscaped areas (eg. sound barriers), cliffs and other sharp features	ignore	View the nDSM sun-shaded. Small ridge lines from dunes and sound barriers can be seen when this error occurs.	If in the coastal plains try surface fitting with the *hills* parameters instead
Boundary issues from piecewise calculation of the GEM	*stitchGEM* ignore	Same as *rfInGEM*. Only occurs where there isn't any data in the gmk.	
DSM has smoothed the edges of tall buildings (causes the height of adjacent objects, particularly vegetation, to appear extremely tall)	ignore	Most noticeable when viewing the vht map with red corresponding to very high (say above $25m$)	
A tall spike in the DSM in clear, flat ground	*spikeDSM* ignore	view the sun-shaded nDSM and compare to the orthophotos	
DSM can have a small drop/discontinuity along small water courses.	ignore	No easy way found to detect these errors. The DSM generation methods may have been improved such that this does not occur.	
Stitch/feathering-like errors in the DSM. Caused by frames that are incorrectly triangulated.	*triangErr* record only if large	In the sun-shaded gem look for consistent height changes that are along straight approximately East-West or North-South lines. Check the orthophotos to see if the jumps are due to a real feature on the ground. Check that height changes are in the dsm too	
A cluster of DSM errors caused by cloud in one of the frames used for stereo matching (cloud not necessarily seen in the orthophotos)	*cloudDSM* repair only if large	View the *nDSM* of the entire tile coloured according to height *(not sun-shaded)*. Anything that is extremely high ($> 100m$), unusually shaped and not due to a poor gem is probably a DSM error	set the DSM to NULL using thresholds or vectors, possibly with the help of a distance transform
GEM completely missed medium sized hills/peaks due to large pieces of clear ground omitted in the ground mask	*peaksMissedGmkGEM* repair if large	View the *nDSM* of the entire tile coloured according to height *(not sun-shaded)*. Compare anything that is extremely high ($> 100m$) with the DSM to see if entire sections hills have been missed	set the GEM to NULL
Low outliers in the DSM that have persisted into the GEM	*belGrdGEM* record only		See section 4.1.3
corrupt DSM heights (caused by read/write error)	*corruptDSM* repair		set the DSM to NULL, and if appropriate set the GEM and digital orthomosaick to NULL too.

Actions: *ignore* = do nothing. *record*= create a brief, rough vector in the unfixed-error vector file. *repair* = record and repair error.

Table E.1: Known issues with heights and any solutions.

Issue	*Keyword* **/Action**	**Displays/Techniques Used for Manual Detection**	**Repair**
A boundary between image frames that was noticeable in the vegetation mask	*frmBndInVeg* ignore /record	Straight boundaries between differing amounts of vegetation density. Can be seen in any display involving the vegetation mask	
A large amount of brown vegetation that was not included in the vegetation mask.	*brVeg* ignore/record	Most obvious when it is bushes/trees that are missed. Best seen with a false colour display with the vegetation mask displayed on top	
Blue roofs in the vegetation mask	*blRfVeg* ignore/record	Same display as *rfInVeg*. Confirm that the roof is blue using the orthophotos.	
Other roofs labelled vegetation	*rfInVeg* record if large amount of roofs	view a 70% transparent vegetation mask (coloured green) on top of a sun-shaded DSM or nDSM	
Peaks, and smooth parts of trees absent from the vegetation mask	*vegInRf* ignore	Caused by tree parts being included in the roof mask. Same view as *rfInVeg*	
Shadows, especially negative spectral values labelled as vegetation	*shInVeg* ignore /record if large amount	Can sometimes be detected using the *rfInVeg* due to the shape of the vegetation mask. Also found with the orthophotos overlayed by the vegetation mask	
A cloud shadow, especially those that caused errors in the vegetation mask	*cldShadow* record	Look for patches of low nir (using a false colour display), often visible as a dark patch when zoomed out. Check that the vegetation mask really is effected	See section 4.1.3
Water in the vegetation mask	*waterInVeg* record/repair if massive	View using the orthophotos or sunshaded nDSM overlayed by the vegetation mask	Either drew vectors or used a threshold on near-infrared to set water to non-vegetation
Cloud in the orthophotos	*cld* record	Easily seen in the orthophotos as semi-transparent patches of white	
regions of forest where the vegetation mask contained ground or shadow	*forestSpec* record/repair if massive	Often noticeable as forest area that were almost completely labelled vegetation. Due to shadows this should occur very rarely in typical ranges of spectral values.	See section 4.1.3
Large amount of dirt in the vegetation mask (around 20000m^2 or more)	*dirtVeg* record/ repair if massive	Best seen with the vegetation mask displayed over the orthophotos, will probably have sharp edges due to a frame boundary	Either drew vectors or used a threshold on near-infrared to set the dirt to non-vegetation
Wetland/swamps/riparian vegetation omitted from the vegetation mask due to moisture absorbing infrared	*wetlandVeg* ignore/ record if large	view the vegetation mask with the orthophotos.	
Synthetic blue (non-roof) cover marked as vegetation	*blSynthVeg* ignore /record if large amount	Only detectable in a comparison between the orthophotos and the vegetation mask.	
Synthetic green things (such as tennis courts) that are marked as vegetation	*grSynthVeg* ignore /record if large amount	Only detectable in a comparison between the orthophotos and the vegetation mask, often common sense and prior knowledge is required to determine whether courts are natural grass or not.	
Small errors caused by noise in the spectral data	*noiseVeg* ignore		
Regions that contain pixels that should be labelled vegetation, but aren't for an unknown reason	*isVeg* ignore	Different cases of these errors may be seen in various differing displays.	
Terrain, such as steep slopes, that have adversely affected the vegetation mask	*terrainIlluminationVeg* ignore	Visible as a large region of poor vegetation mask on a steep slope and not in a dense forest.	

Actions: *ignore* = do nothing. *record*= create a brief, rough object in the `eru` file. *repair* = record and then fix error.

Table E.2: Known issues with the vegetation masks and any solutions.

Product	**Abbrv**	**Data type**
DSM	dsm	Signed32BitInt. Units are mm above geoid.
candidate ground points	gmk	Signed32BitInt. Units are mm above geoid.
roof mask	rmk	Signed32BitInt. Units are mm above geoid.
GEM	gem	Signed32BitInt. Units are mm above geoid.
nDSM	nsm	Signed32BitInt. Units are mm above ground.
Multispectral	dom	Signed16BitInt. Ground reflectance. Band 1=Red, Band 2=Green, Band 3=Blue, Band 4=nir
Vegetation/non-vegetation classification	veg	Unsigned8BitInt. 1=Veg,NULL=nonveg=0
Vegetation height	vht	Signed32BitInt NULL=Non-veg
Vegetation index	vin	IEEE4ByteReal NULL=-320000=non Veg
Green Grass Mask	grs	Unsigned8BitInt. Green grass =1, NULL=non-grass=0
Bushes/trees Mask	tre	Unsigned8BitInt. Bush/tree = 1, Null=non-bush/tree=0
No data mask	msk_nod	Unsigned8BitInt. No dom =1, no dsm/gem/ndsm =2 neither =3

Table E.3: The data sets created over the full Urban Monitor region for 2007 and 2009. The abbreviations were used in the file naming conventions (see [13, section 4]).

roof mask

1. `winflows` was applied to the nDSM with a slope threshold of 25^0, an area threshold of 1000 pixels and elevation threshold of $1m$.

2. All the of the segments with values greater than 0 and less than 51 were used to create the roof mask.[3]

Vegetation Classification

1. For 2007 `returnVegMask` v1.3 was run with the multispectral data and roof mask as inputs. In later version of `returnVegMask` this was equivalent to using the parameters $nirO = 0$, $rO = 0$, $nirErr = 500$, $rErr = 500$, a vegetation threshold of 0.2 and a roof cleaning threshold of 0.3.

2. For 2009 `returnVegMask` (v1.4) was run with $nirO = -50$, $rO = 75$, $nirErr = 100$, $rErr = 100$, a vegetation threshold of 0.2 and a roof cleaning threshold of 0.3. Data inputs were the multispectral mosaic and the roof mask.

Vegetation Heights

1. `vegHeights` was run with the nDSM and the vegetation mask as inputs.

Vegetation Index

1. `vegNDVI` was run with the multispectral mosaick and vegetation mask as inputs.

Green grass mask or bushes/trees mask

1. `vegBetweenHeights` was run with heights between -1000 and 500 or 500 and 1000000 respectively.

No data mask

1. `noDataMask` was run with the multispectral data and the nDSM as inputs.

[3]The creation and feathering of the 20000×20000 pieces is not described here. For the full region these steps were carried out by Drew Devereux.

E.3 Description of programs: what they do, and how to use them

Inputs of different sizes: The supplied programs generated outputs with extents that were the intersection of the extents of the inputs. For example if a small subset of the multispectral data was extracted and then `returnVegMask` run on a larger roof mask file then a file the same size as the smaller region was created. The exceptions to this were `simplePaste` and `featherPaste`.

Help: If the incorrect number of arguments or invalid options were entered then a brief description of the program and instructions on how to use it were displayed.

Automatically Generating History Files: Most programs contained options to generate associated history files. These caused the programs to copy the history files of the input data into the output data's history file. The options were either "-buildTempHistory" or "-buildHistory". In the case of simplePaste and featherPaste, only one option "-addToHistory" was available, and it caused the programs to append a line to the most recent history file.

applyMaskToSurface

This program required an input masking image, an input surface, an output file name, a lower mask bound and an upper mask bound.

The output was the surface except where the masking image values were *outside* the bounds given (either strictly lower than the lower mask bound or strictly greater than the upper mask bound).

candGrdForGEMPatch

Input data was a GEM, a ground mask (gmk) and a class file. It also required an out file name.

The program created a set elevation points (designed to be control points for surface fitting) where the class file specified where to use heights from the GEM, the gmk, or neither. It assumed that the values of the class file corresponded to

- 0=Use neither - pixel will be NULL

- 1=Use the ground mask - pixel will take the value of the ground mask at the same location
- 2=Use the ground elevation surface - pixel will take the value of the gem at the same location

It also had an option '-writeGMKPatch=[out patch name]' to export a ground mask suitable for repairing the gmk if desired.

candGrdForGEMPatchwCleaning

Input raster data was a gem, a ground mask, a class file, and a reference GEM (often of another year). It also required an error threshold for cleaning and an out candidate ground image.

The program created a set elevation points (designed to be control points for surface fitting) where the class file specified where to use heights from the gem, the ground mask, a cleaned version of the ground mask, or use no height data. It assumed that the values of the class file correspond to

- 0=Use neither - pixel will be NULL
- 1=Use the ground mask - pixel will take the value of the ground mask at the same location
- 2=Use the GEM - pixel will take the value of the GEM at the same location
- 3=Use height data from the ground mask, but remove any points whose difference to the reference GEM are greater than the error threshold.

It also had an option '-writeGMKPatch=[out patch name]' to export a ground mask suitable for repairing the gmk if desired.

featherPaste

Input was a main image, a piece image and a feather distance (in metres).

Created a distance transform image using some of Peter Caccetta's code. This estimated the distance of each pixel to the closest NULL value or the image boundary. It then feathered the data into the main image weighting each according to the distance.

The distance transform assumed that the NULL value was by far the lowest value in the image. The feather distance was roughly converted to the units of the distance transform (which was not in metres). Maximum feathering distance was about $66m$. A feather distance of 'NONE' pasted the data including all the NULL values without any feathering.

This program only worked for Signed32BitIntegers.

GEMfix.bat

This batch script combined a rasterizing program[4] with `candGrdForGEMPatch` and `Terzopoulos` to generate repaired GEM.

Inputs were an ERMapper vector file, an input GEM, an input gmk and the parameter set to use for `Terzopoulos`.

The rasterizing program created a class image according to the names of the polygons in the ERMapper vector file.

- if the pixel was inside a polygon labelled 'use_gmk' then it was given a value of 1 in the class file.
- Otherwise if the pixel was inside a polygon labelled 'del' then it was given a value of 0 in the class file.
- Otherwise pixels were given a value of 2 in the class file.

This class file was then passed to `candGrdForGEMPatch` along with the input GEM and gmk. Finally `tpsToGriddedPoints` was run on the resulting set of control points.

GEMFixWCleaning.bat

This batch script combined a rasterizing program[5] with `candGrdForGEMPatch`, `Terzopoulos` and a reference GEM to generate repaired GEMs.

Inputs were an ERMapper vector file, an input GEM, an input gmk, a reference GEM, an error threshold and the parameter set to use for `Terzopoulos`.

The rasterizing program created a class image according to the names of the polygons in the ERMapper vector file:

[4]written by Drew Devereux

[5]written by Drew Devereux

- if the pixel was inside a polygon labelled 'use_gmk' then it was given a value of 1.
- Otherwise if the pixel was inside a polygon labelled 'use_cleaned_gmk' then it was given a value of 3.
- Otherwise if the pixel was inside a polygon labelled 'del' then it was given a value of 0.
- Otherwise pixels were given a value of 2.

This class file was then passed to `candGrdForGEMPatchwCleaning` along with the input GEM, input gmk, the reference GEM and the error threshold. Finally `tpsToGriddedPoints` was run on the resulting set of control points.

nDSMfromGEMnDSM

Input was a GEM, a DSM and an out nDSM file name.

Calculated $DSM - GEM$ for all pixels. Wherever $DSM - GEM < 0$ the final value was set to 0.

Only worked for Signed32BitIntegers.

noDataMask

Input was a multispectral image, the nDSM, and an output file name

When all the bands of the multispectral data were NULL and the nDSM was NULL as well then the output value was 3. If only the nDSM was NULL then the output was 2. If only the multispectral bands were NULL then the output value was 1 (note that all the multispectral bands had to be NULL).

returnVegMask

For version 1.4: Input raster data was a multispectral file and a roof mask (in Unsigned8BitIntegers, or Signed32BitIntegers). Other inputs were a near-infrared origin (O_{nir}), an origin for the red band (O_r), an error bound on the near-infrared data (tol_{nir}), an error bound on the red band (tol_r), a threshold for the NDVI above which pixels were labelled vegetation ($vegThreshold$) and a

threshold below which vegetation pixels were relabelled depending on the value roof mask ($rfCleanTh$).

For **2007** the parameters were $O_{nir} = O_r = 0$, $tol_{nir} = tol_r = 500$, $vegThreshold = 0.2$ and $rfCleanTh = 0.3$ (this matched the results of earlier versions of `returnVegMask`).

For **2009** the parameters were $O_{nir} = -50$, $O_r = 75$, $tol_{nir} = tol_r = 100$, $vegThreshold = 0.2$ and $rfCleanTh = 0.3$.

For each pixel `returnVegMask` calculated:

1. if the near infrared band (nir) or red band (r) were NULL then the pixel was labelled NULL too.

2. calculated NDVI based upon the formula given in section 3.2.

3. if $ndvi > rfCleanTh$ then the pixel was set to vegetation.

4. if $rfCleanTh > ndvi > vegThreshold$ and the pixel was *not* roof then it was also set to vegetation.

Returned 1 if a pixel was vegetation, 0 if not vegetation, or NULL (in the Urban Monitor data NULL was 0 so non-vegetation was not distinguishable from dark pixels). Output was an Unsigned8BitInteger.

return2007VegMaskWeights

Input a multispectral image, and an output file name.

This program was created for a single purpose: to generate a feathering distance to fix the vegetation mask in the hills. It calculated the distance (in metres) from a hard coded polyline running roughly North-South along the hills (a vector file for it can be found in `vegHillsEdge.erv`). The Eastern side of the polyline was given a value of 0. It didn't actually read the information in the multispectral image, but simply used its extents to generate a distance map.

returnVegMaskwHills

This program was used to feather the parameters for the Darling Ranges forest with the normal vegetation parameters for the 2007 data.

Input was a multispectral image, the roof mask, the distance file (created using `return2007VegMaskWeights`) and an output file.

The thresholds on NDVI remained the same as `returnVegMask` (0.2 and 0.3). However the parameters $O_{nir}, O_r, tol_{nir}, tol_r$ changed linearly according to the distance map. Inside the hills the parameters were those given in section 4.1.3 and 4000m away from the boundary the parameter values matched the rest of the Urban Monitor region. The vegetation label was calculated using the same method in `returnVegMask`.

Returned 1 if the pixel was vegetation, 0 if not vegetation, or NULL (in the Urban Monitor data NULL was 0 so non-vegetation was not distinguishable from dark pixels). Output was an Unsigned8BitInteger.

scanForDifference

Required two input files, A and B.

Scanned the overlap between A and B and printed the total number of pixels that were different, the largest difference, and the Easting-Northing of the first time this largest difference was encountered. Worked for multiple data types. (Note there may be some issues with this program. In some comparisons it recorded large differences, but at the Easting-Northing it returned no differences were apparent)

simplePaste

Input was main image, piece image and a mask value.

Worked for Unsigned8BitInteger, Signed16BitInteger, Signed32BitInteger, IEEE4ByteReal, IEEE8ByteReal. Overwrote the data in the main image with the values in the piece image without any feathering.

It ignored (does not copy) the mask value. If NONE was used here then it copied the data *verbatim. Warning: by default the program copied NULL values too.*

This program still functioned if the piece image was not a subset of the main image. It only copied the region that both images had in common.

vegNDVI

Inputs were a multispectral file and a vegetation mask, and an out file name.

This program did not use data offsets or tolerances. If $nir < 0$ then it returned -1, if $r < 0$ then it returned $+1$, otherwise it calculated $(nir - r)/(nir + r)$ for every pixel in the vegetation mask. Because the vegetation masks to date had not allowed pixels with both $nir < 0$ and $r < 0$ the program did not have a rule for dealing with them: a pixel with both $nir < 0$ and $r < 0$ would have been given a value of 1.

Output was IEEE4ByteReals.

vegHeights

Inputs were an nDSM, a vegetation mask and an out file name.

Simply returned the nDSM value at every pixel in the vegetation mask. Output was in Signed32BitInteger

vegBetweenHeights

Input was the vegetation heights map, a lower and upper height bound, and an out file name.

Returned a 1 for all pixels in the map such that $lowBound < height <= upperBound$. Output data type was Unsigned8BitInteger

winflows

This was an implementation of the inflows algorithm written by Peter Caccetta (described in section 2.2, appendix C.1 and [12]) that worked on large datasets. Input was a surface model (e.g. DSM, nDSM, GEM), an output file name, a scale factor, a slope threshold, an area threshold and an elevation threshold.

Bibliography

[1] Nizar Abo Akel, Ofer Zilberstein, Yerach Doytsher, Geo-information Engineering, Technion City, Systems Group, and Global Position. A robust method used with orthogolan polynomials and road network for automatic terrain surface extraction from lidar and urban areas. In Orhan Altan, editor, *Proceedings of the XXth ISPRS Congress, Technical Commission III*, pages 243–248, Istanbul, Turkey, 2004.

[2] Peter Axelsson. Processing of laser scanner data - algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3):138–147, July 1999.

[3] Peter Axelsson. DEM generation from laser scanner data using adaptive TIN models. *International Archives of Photogrammetry and Remote Sensing*, XXXIII:110–117, 2000.

[4] C Baillard. A hybrid method for deriving DTMs from urban DEMs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 29(3):21, 2008.

[5] E. P. Baltsavias. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(3-4):129–151, January 2004.

[6] E.P. Baltsavias. *Airborne laser scanning: existing systems and firms and other resources*, volume 54. July 1999.

[7] A Bannari, D Morin, F Bonn, and A R Huete. A review of vegetation indices. *International Journal of Remote Sensing*, 13(1-2):95–120, 1995.

[8] S. Bedawi and M. Kamel. Segmentation of Very High Resolution Remote Sensing Imagery of Urban Areas Using Particle Swarm Optimization Algorithm. *Image Analysis and Recognition*, pages 81–88, 2010.

[9] Thomas Blaschke, Kasper Johansen, and Dirk Tiede. Object-Based Image Analysis for Vegetation Mapping and Monitoring. In Qihao Weng, editor, *Advances in Environmental Remote Sensing: Sensors, Algorithms, and Applications*, Remote Sensing Applications Series, chapter 10, pages 241–272. CRC Press, 2011.

[10] P. Boccardo and G. Gentili. High resolution DSM and classified volumetric generation: an operational approach to the improvement of geospatial intelligence. In *ISPRS Hannover Workshop 2011 High-Resolution Earth Imaging for Geospatial Information*, Hannover, 2011.

[11] Matthew Bolitho, Michael Kazhdan, Randal Burns, and Hugues Hoppe. Multilevel streaming for out-of-core surface reconstruction. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 69–78. Eurographics Association, 2007.

[12] Peter Caccetta. *Remote sensing, geographic information systems (GIS) and Bayesian knowledge-based methods for monitoring land condition*. Thesis, Curtin University of Technology, 1997.

[13] Peter Caccetta, Simon Collings, Andrew Devereux, Kass Hingee, Don Mcfarlane, Anthony Traylen, and Xiaoliang Wu. Urban Monitor: Enabling effective monitoring and management of urban and coastal environments using digital aerial photography; Final Report - Transformation of aerial photography into digital raster information products. Technical Report October, CSIRO, Perth, 2012.

[14] Peter Caccetta, Robert Dunne, Richard George, and Don McFarlane. A methodology to estimate the future extent of dryland salinity in the southwest of Western Australia. *Journal of environmental quality*, 39(1):26–34, 2005.

[15] Peter A Caccetta, Simon Collings, Kass Hingee, Don McFarlane, and Xiaoliang Wu. Fine-scale monitoring of complex environments using remotely

sensed aerial , satellite , and other spatial data. In *2011 International Symposium on Image and Data Fusion (ISIDF)*. IEEE, 2011.

[16] Peter A Caccetta, D Mcfarlane, Xiaoliang Wu, and Simon Collings. Urban Monitor - fine scale monitoring of complex environments. In *7th International Symposium on Digital Earth*, number August, page 8. International Society for Digital Earth, August 2011.

[17] Norm A Campbell. Shrunken estimators in discriminant and canonical variate analysis. *Applied Statistics*, 29(1):5–24, 1980.

[18] Norm A Campbell. Robust procedures in multivariate analysis II. Robust canonical variate analysis. *Applied Statistics*, 31(1):1–8, 1982.

[19] Norm A Campbell and William R. Atchley. Geometry of Canonical Variate Analysis. *Systematic zoology*, 30(3):268–280, 1981.

[20] Norm A Campbell and Richard Reyment. Discriminant analysis of a Cretaceous foraminifer using shrunken estimators. *Mathematical Geology*, 10(4):347–359, 1978.

[21] Nicolas Champion and Didier Boldo. A robust algorithm for estimating digital terrain models from digital surface models in dense urban areas. In *Proceedings ISPRS Commission 3 Symposium, Photogrammetric Computer Vision*, 2006.

[22] Nicolas Champion, Didier Boldo, Marc Pierrot-Desilligny, and Georges Stamon. Automatic estimation of fine terrain models from multiple high-resolution satellite images. In *16th IEEE International Conference on Image Processing (ICIP)*, pages 577–580, 2009.

[23] Nicolas Champion, Franz Rottensteiner, Leena Matikainen, Xinlian Liang, J. Hyyppä, and BP Olson. A test of automatic building change detection approaches. *Proceedings of CMRT09*, XXXVIII:03–04, 2009.

[24] Ya-Mei Chang, Adrian Baddeley, Jeremy F Wallace, and Michael Canci. Spatial statistical analysis of changes in native vegetation using airborne

digital imagery. In *JSM Proceedings: Section on Statistics and the Environment*, volume 124, Miami Beach, Florida, 2011. American Statistical Association.

[25] Nesrine Chehata and Frederic Bretar. Terrain modeling from LIDAR data: Hierarchical K-means filtering and markovian regularization. In *IEEE International Conference on Image Processing*, pages 1900–1903, 2008.

[26] Qi Chen, Dennis Baldocchi, Peng Gong, and Maggi Kelly. Isolating Individual Trees in a Savanna Woodland Using Small Footprint Lidar Data. *Photogrammetric Engineering & Remote Sensing*, 72(8):923–932, 2006.

[27] Xuehong Chen, Jin Chen, Yusheng Shi, and Yasushi Yamaguchi. An automated approach for updating land cover maps based on integrated change detection and classification methods. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:86–95, July 2012.

[28] Joanne L. C. Chia, Norm A Campbell, and Peter A Caccetta. Using Rational Polynomials for Mapping in Areas with Terrain without Using a Digital Elevation Model. In *12th Australasian Remote Sensing and Photogrammetry Conference Proceedings*, October 2004.

[29] Simon Collings, Peter A Caccetta, Norm A Campbell, and Xiaoliang Wu. Empirical Models for Radiometric Calibration of Digital Aerial Frame Mosaics. *IEEE Transactions on Geoscience and Remote Sensing*, 49(99):1–16, July 2011.

[30] RG Congalton. A review of assessing the accuracy of classifications of remotely sensed data. *Remote sensing of environment*, 46(October 1990):35–46, 1991.

[31] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin. Digital change detection methods in ecosystem monitoring: a review. *International Journal of Remote Sensing*, 25(9):1565–1596, May 2004.

[32] Robert E. Crippen. The dangers of underestimating the importance of data adjustments in band ratioing. *International Journal of Remote Sensing*, p(4):767–776, 1988.

[33] W.A. Davidson. *Hydrogeology and groundwater resources of the Perth region, Western Australia*. Geological Survey of Western Australia, Perth, 1995.

[34] N Demir, D Poli, and E. P. Baltsavias. Extraction of buildings and trees using images and Lidar data. *ISPRS Proceedings, Technical Commission IV, Beijing*, pages 313–318, 2008.

[35] A. DomaÇ and M. L. Süzen. Integration of environmental variables with satellite images in regional scale vegetation classification. *International Journal of Remote Sensing*, 27(7):1329–1350, April 2006.

[36] CC Douglas. Multigrid methods in science and engineering. *IEEE Computational Science & Engineering*, 3(4):55–68, 1996.

[37] W Eckstein and O Munkelt. Extracting objects from digital terrain models. In *Remote Sensing and Reconstruction for Three-Dimensional Objects and Scenes, SPIE*, pages 43 – 51, 1995.

[38] R. J. Fensham and R. J. Fairfax. Aerial photography for assessing vegetation change: a review of applications and the relevance of findings for Australian vegatation history. *Australian Journal of Botany*, 50:415–429, 2002.

[39] Giles M. Foody. Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, 80(1):201–185, 2002.

[40] Suzanne Furby, Peter Caccetta, and Jeremy Wallace. Salinity monitoring in Western Australia using remotely sensed and other spatial data. *Journal of environmental quality*, 39(1):16–25, 2010.

[41] Suzanne L Furby. *Land Cover Change: Specification for Remote Sensing Analysis*. Australian Greenhouse Office, Canberra, 2002.

[42] Muhittin Gokmen and Ching-Chung Li. Edge detection and surface reconstruction using refined regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):492–499, May 1993.

[43] T. K. Gottschalk, F. Heuttmann, and M. Ehlers. Thirty years of analysing and modelling avian habitat relationships using satellite imagery data: a review. *International Journal of Remote Sensing*, 26(12):2631–2656, 2005.

[44] Nancy Hoalst-Pullen and Mark W Patterson. Applications and Trends of Remote Sensing in Professional Urban Planning. *Geography Compass*, 5(5):249–261, 2011.

[45] RJ Hobbs, Jeremy F Wallace, and Norm A Campbell. Classification of vegetation in the Western Australian wheatbelt using Landsat MSS data. *Plant Ecology*, 80(2):91–105, 1989.

[46] Joachim Höhle and Marketa Potuckova. Assessment of the Quality of Digital Terrain Models. Technical Report 60, European Spatial Data Research, 2011.

[47] Brent Holben and Chris Justice. An examination of spectral band ratioing to reduce the topographic effect on remotely sensed data. *International Journal of Remote Sensing*, 2(2):115–133, 1981.

[48] Xin Huang and Liangpei Zhang. Classification and extraction of spatial features in urban areas using high-resolution multispectral imagery. *Geoscience and Remote Sensing*, 4(2):260–264, April 2007.

[49] John R. Jensen and D.C. Cowen. Remote sensing of urban/suburban infrastructure and socio-economic attributes. *Photogrammetric Engineering and Remote Sensing*, 65:611–622, 1999.

[50] Harri Kaartinen, Juha Hyyppä, E Gülch, George Vosselman, H Hyyppä, Leena Matikainen, AD Hofmann, U Mäder, ÅPersson, U. Söderman, M. Elmqvist, A. Ruiz, M. Dragoja, D. Flamanc, G. Maillet, T. Kersten, J. Carl, R. Hau, E. Wild, L. Frederiksen, J. Holmgaard, and K. Vester. Accuracy of 3D city models: EuroSDR comparison. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W19):227–232, 2005.

[51] Stefan Kluckner. *Semantic Interpretation of Digital Aerial Images Utilizing Redundancy, Appearance and 3D Information.* Phd thesis, Graz University of Technology, February 2011.

[52] Stefan Kluckner, Thomas Mauthner, P. Roth, and Horst Bischof. Semantic classification in aerial imagery by integrating appearance and height information. *Computer Vision-ACCV 2009*, pages 477–488, 2010.

[53] Joseph F Knight, Ross S Lunetta, Jayantha Ediriwickrema, and Siamak Khorram. Regional Scale Land Cover Characterization Using MODIS-NDVI 250 m Multi-Temporal Imagery: A Phenology-Based Approach. *GI-Science Remote Sensing*, 43(1):1–23, 2006.

[54] Andrej Kobler, Norbert Pfeifer, Peter Ogrinc, Ljupčo Todorovski, Krištof Oštir, and Sašo Džeroski. Repetitive interpolation: A robust algorithm for DTM generation from Aerial Laser Scanner Data in forested terrain. *Remote Sensing of Environment*, 108(1):9–23, May 2007.

[55] Florent Lafarge and Clément Mallet. Creating Large-Scale City Models from 3D-Point Clouds: A Robust Approach with Hybrid Representation. *International Journal of Computer Vision*, 99(1):69–85, February 2012.

[56] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, 9(1):112, 1998.

[57] F. Leberl and M. Gruber. Flying the new large format digital aerial camera Ultracam. In D. Fritsch and Ed. Heidelberg, editors, *Photogrammetric Week*, pages 67–76. Wichmann Verlag, 2003.

[58] Jose Miguel Leiva-Murillo and Antonio Artés-Rodríguez. Maximization of mutual information for supervised linear feature extraction. *IEEE transactions on neural networks*, 18(5):1433–41, September 2007.

[59] Zhilin Li, Qing Zhu, and Christopher Gold. *Digital terrain modeling: principles and methodology.* CRC, 2005.

[60] Jiping Liu, Jing Shen, Rong Zhao, and Shenghua Xu. Extraction of individual tree crowns from airborne LiDAR data in human settlements. *Mathematical and Computer Modelling*, November 2011.

[61] Xiaoye Liu. Airborne LiDAR for DEM generation: some critical issues. *Progress in Physical Geography*, 32(1):31–49, February 2008.

[62] Karl-ludwig Lothhammer. State-of-the-art photogrammetric workflow the inpho approach. *Semana Geomatica*, pages 1–9, 2005.

[63] D. Lu, P. Mausel, E. Brondízio, and E. Moran. Change detection techniques. *International Journal of Remote Sensing*, 25(12):2365–2401, June 2004.

[64] D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870, March 2007.

[65] Leena Matikainen, Harri Kaartinen, and Juha Hyyppä. Classification tree based building detection from laser scanner and aerial image data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3):280–287, 2007.

[66] SK McFeeters. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *International journal of remote sensing*, pages 37–41, 1996.

[67] Domen Mongus and Borut Žalik. Parameter-free ground filtering of LiDAR data for automatic DTM generation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67:1–12, January 2012.

[68] J A Nelder and R Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 1965.

[69] Trisalyn Nelson, Barry Boots, and Michael a Wulder. Techniques for accuracy assessment of tree locations extracted from remotely sensed imagery. *Journal of environmental management*, 74(3):265–71, February 2005.

[70] Janet Elizabeth Nichol, Man Sing Wong, Richard Corlett, and Douglas W. Nichol. Assessing avian habitat fragmentation in urban areas of Hong Kong (Kowloon) at high spatial resolution using spectral unmixing. *Landscape and Urban Planning*, 95(1-2):54–60, March 2010.

[71] Norbert Pfeifer, Martin Rutzinger, Franz Rottensteiner, Werner Muecke, and Markus Hollaus. Extraction of building footprints from airborne laser scanning: Comparison and validation techniques. *2007 Urban Remote Sensing Joint Event*, pages 1–9, April 2007.

[72] C. Pohl and JL Van Genderen. Review article Multisensor image fusion in remote sensing: concepts, methods and applications. *International journal of remote sensing*, 19(5):823–854, 1998.

[73] R Development Core Team. *R: A Language and Environment for Statistical Computing: Reference Index*. R Foundation for Statistical Computing, 2.13.0 edition, 2011.

[74] Richard J Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms: a systematic survey. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 14(3):294–307, March 2005.

[75] K Reinke and Simon Jones. Integrating vegetation field surveys with remotely sensed data. *Ecological Management & Restoration*, 7(June):18–23, 2006.

[76] David Riaño, Emilio Chuvieco, Javier Salas, and Inmaculada Aguado. Assessment of Different Topographic Corrections in Landsat-TM Data for Mapping Vegetation Types (2003). *IEEE Transactions on Control*, 41(5):1056–1061, 2003.

[77] M.K. Ridd and Jiajun Liu. A comparison of four algorithms for change detection in an urban environment. *Remote sensing of environment*, 63(2):95–100, 1998.

[78] J. W. Rouse, R. H. Haas, J. A. Shcell, and D. W. Deering. Monitoring vegetation systems in the great plains with ERTS. In *NASA. Goddard Space Flight Center 3 d ERTS-1 Symp,*, 1974.

[79] Osvaldo E. Sala. Section 5. In H. A. Mooney, J. Lubchenco, R. Dirzo, and O. E. Sala, editors, *Global Biodiversity Assessment*. Cambridge Univ. Press, Cambridge, 1995.

[80] Osvaldo E. Sala, F. Stuart III Chapin, Juan J. Armesto, Eric Berlow, Janine Bloomfield, Rodolfo Dirzo, Elisabeth Huber-Sanwald, Laura F. Huenneke, Robert B. Jackson, Ann Kinzig, Rik Leemans, David M. Lodge, Harold A. Mooney, Martin Oesterheld, N. LeRoy Poff, Martin T. Sykes, Brian H. Walker, Marilyn Walker, and Diana H. Wall. Global Biodiversity Scenarios for the Year 2100. *Science*, pages 1770–1774, March 2000.

[81] A.K. Shackelford and C.H. Davis. A fuzzy classification approach for high-resolution multispectral data over urban areas. In *IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages 1621–1623. IEEE, 2002.

[82] Ashbindu Singh. Review Article Digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, 10(6):989–1003, 1989.

[83] George Sithole. Filtering of laser altimetry data using a slope adaptive filter. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34:203–2010, 2001.

[84] George Sithole and George Vosselman. Report. ISPRS Comparsion of Filters. Technical Report 3, Internation Society for Photogrammetry and Remote Sensing, February 2003.

[85] George Sithole and George Vosselman. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(1-2):85–101, August 2004.

[86] Terrence E Slonecker, David B Jennings, and Donald Garofalo. Remote sensing of impervious surfaces : A review. *International Journal of Remote Sensing*, 20(3):227–255, 2001.

[87] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513–528, June 1990.

[88] H. Taubenbock and T. Esch. Remote Sensing - An Effective Data Source for Urban Monitoring, 2011.

[89] JC Taylor, TR Brewer, and AC Bird. Monitoring landscape change in the national parks of England and Wales using aerial photo interpretation and GIS. *International Journal of Remote Sensing*, 21(13 - 14):27372752, 2000.

[90] Demetri Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 24(1):52–96, 1983.

[91] Demetri Terzopoulos. *Multiresolution computation of visible-surface representations*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, July 1984.

[92] Demetri Terzopoulos. The computation of visible-surface representations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(4):417–438, July 1988.

[93] D Tóvári and Norbert Pfeifer. Segmentation based robust interpolation-a new approach to laser data filtering. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W19):79–84, 2005.

[94] Roger Trias-Sanz, Georges Stamon, and Jean Louchet. Using colour, texture, and hierarchial segmentation for high-resolution remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):156–168, March 2008.

[95] Te-ming Tu, Chin-hsing Chen, Jiunn-lin Wu, and Chein-i Chang. A Fast Two-Stage Classification Method for High-Dimensional Remote Sensing

Data. *IEEE Transactions on Geoscience and Remote Sensing*, 36(1):182–191, 1998.

[96] United Nations. World Urbanization Prospects, The 2011 Revision: Highlights. Technical report, United Nations, New York, New York, USA, 2012.

[97] Rickard van Dongen. Vegetation structure mapping with Urban Monitor Data for the Yeal Nature Reserve. In *Proceedings of the Internation Geoscience and Remote Sensing Symposium*. (submitted), July 2013.

[98] PM Vitousek. Beyond global warming: ecology and global change. *Ecology*, 75(7):1861–1876, 1994.

[99] Douglas Ward, Stuart R Phinn, and Alan T Murray. Monitoring Growth in Rapidly Urbanizing Areas. *Professional Geographer*, 52(3):371–386, 2000.

[100] Western Australian Planning Commission. Capital City Planning Framework: a vision for Central Perth. Technical Report February, Western Australian Planning Commission, Perth, 2013.

[101] Y. Xie, Z. Sha, and M. Yu. Remote sensing imagery in vegetation mapping: a review. *Journal of Plant Ecology*, 1(1):9–23, April 2008.

[102] L Zebedin, A Klaus, B Grubergeymayer, and K Karner. Towards 3D map generation from digital aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(6):413–427, September 2006.

[103] Y Zha, J Gao, and S Ni. Use of normalized difference built-up index in automatically mapping urban areas from TM imagery. *International Journal of Remote Sensing*, 24(3):583–594, 2003.

[104] Haohao Zhao, Xuezhi Feng, and Shuhe Zhao. The application of entropy in the urban green information extraction of high resolution remote sensing. In *International Conference on Information Science and Technology*, pages 604–607. IEEE, March 2011.

[105] S Zoraster. A surface modeling algorithm designed for speed and ease of use with all petroleum industry data. *Computers & Geosciences*, 29(9):1175–1182, November 2003.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.